

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en ingeniería informática

TRABAJO FIN DE GRADO

Módulo de control de iluminación en Unity a partir de patrones musicales

Adrián Morales Montón

Tutor: Carlos Aguirre Maeso

Ponente: Adrián Morales Montón

JULIO 2019

Módulo de control de iluminación en Unity a partir de patrones musicales

AUTOR: Adrián Morales Montón

TUTOR: Carlos Aguirre Maeso

**Dpto. Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2019**

Resumen (castellano)

Gracias al auge de los juegos independientes, en los que se valora altamente la originalidad y la calidad artística del producto final, la música en los videojuegos, se ha transformado en un pilar esencial en el diseño, en el que incluso, en ocasiones, llega a superar al apartado visual. Por esta razón se ha decidido la realización de este TFG, el cual podrá aprovechar este impulso y aportar algo nuevo que mejore aún más esta característica.

Este Trabajo Fin de Grado consiste en el análisis de la música para crear un entorno o escena dinámica que responda a los estímulos musicales entrantes y que se transforme y cambie según la emoción o sentimiento que genere una canción.

Primeramente, se diseñará y crearán una serie de scripts capaces de analizar las diferentes emociones transmitidas por una obra musical concreta. Para ello habrá que estudiar qué características las definen y como identificarlas. Por otro lado, para el desarrollo de la solución planteada, se hará uso de la transformada de Fourier (así como de sus variantes), funciones ventana, conocimientos sobre armónicos y ritmo para construir mediante múltiples variables (amplitud, frecuencia ...) una forma óptima de analizar cada pieza musical e interpretarla correctamente.

El resultado de este TFG será orientado para la creación de una serie de respuestas o efectos (niebla, mayor o menor luminosidad, lluvia, etc.) que responderán a los datos analizados por el algoritmo mencionado anteriormente, consiguiendo así una escena dinámica y cambiante que este en sinergia con la pieza musical que se reproduzca en ese momento.

Este trabajo se realizará en el motor de videojuegos Unity3D, mediante el código de programación C# y haremos uso de diferentes herramientas externas que nos ayudaran a el correcto análisis de señales de audio.

Abstract (English)

Thanks to the peak of indie games in which originality and artistic quality are highly valued, music in videogames has become a central pillar in its designing, sometimes even surpassing graphics. For this reason, it has been decided to present this TFG, which will be able to take advantage of this, and contribute with something new that will further improve this feature.

This Final Grade Work consists on analyzing music in order to create a dynamic surrounding or scene that responds to entering musical stimulus and goes on transforming itself and changing according to the emotion or feeling that a specific song generates.

First of all, a series of scripts will be designed and created that can analyze the range of different emotions transmitted by a determined song. For this purpose, it will be necessary to study which characteristics define them and how to identify them. On the other side, for the development of the solution, the Fourier transform -as well as its variants-, window functions and also knowledge about harmonics and rhythm are going to be used to build an optimal way of analyzing each musical piece and interpreting it correctly by using a set of variables as amplitude or frequency.

In the second part of this TFG a series of answers or effects as fog, higher or lower luminosity, rain... responding to the data analyzed by the previously mentioned algorithm

will be made, thus obtaining a dynamic and changing scene in synergy with the musical piece played at that moment.

This work will be conducted with the videogame's engine Unity3D by the programming code C# and we will make use of a set of different external tools that will help us to get the right analysis of the audio signals.

Palabras clave (castellano)

Unity, videojuegos, música, patrones, armónicos, iluminación, ritmo, frecuencias, amplitudes.

Keywords (inglés)

Unity, video games, music, patterns, harmonics, lighting, rhythm, frequencies, amplitudes.

Agradecimientos

Me gustaría dedicar estas líneas a mis padres, por darme apoyo todos estos años, y a mi pareja Elsa, por tener la paciencia de estar conmigo siempre en los momentos difíciles.

INDICE DE CONTENIDOS

1. Introducción.....	1
1.1 Motivación.....	1
1.2 Objetivos.....	2
1.3 Organización de la memoria.....	3
2 Estado del arte	4
2.1 Videojuegos.....	4
2.1.1 SingStar	4
2.1.2 AudioSurf	4
2.1.3 Patapon	5
2.1.4 Guitar Hero 2	5
2.2 Software para análisis de audio	5
2.2.1 Audacity	5
2.2.2 Adobe Audition CC	5
2.2.3 SATLive	6
2.2.4 Smaart.....	6
2.3 Transformada de Fourier	6
2.4 DSPLib	9
2.5 Psicología de la música	10
3 Diseño.....	11
3.1 Controlador_fourier	13
3.2 Analizadores	14
3.2.1 Ritmo	16
3.2.2 Tendencia de Frecuencia	16
3.2.3 Silencios	16
3.2.4 Tendencia de intensidad	16
3.3 Contador	17
3.4 Controlador_resultados.....	18
3.5 Diagrama de flujo	19
4 Desarrollo	20
4.1 Controlador_fourier	20
4.2 Analizador: Ritmo	21
4.3 Preprocesamiento de señales de audio – Controlador_fourier	25
4.4 Analizador: Tendencia de Frecuencia	29
4.5 Analizador: Silencios y Tendencia de intensidad.....	30
4.6 Contador	31
Event 32	
4.7 Controlador_resultados.....	33
5 Integración, pruebas y resultados	34
6 Conclusiones y trabajo futuro.....	37
6.1 Conclusiones.....	37
6.2 Trabajo futuro	37
7 Referencias	38
8 Anexos	40
8.1 Manual de instalación.....	40
8.2 Manual del programador	40
8.3 Estudio sobre la música	40
8.4 Representación gráfica de una señal de audio	58

8.4.1 Dominio frecuencial	58
8.4.2 Dominio temporal.....	60
8.4.3 Espectrograma o sonograma.....	60
8.5 Conocimientos previos al diseño.....	63
8.5.1 La música.....	63
8.5.2 El sonido	63
8.5.3 Frecuencia.....	64
8.5.4 Amplitud.....	65
8.5.5 El oído y el sonido	66
8.5.6 Aproximación a la comprensión musical	67

INDICE DE FIGURAS

FIGURA 1 AUDIOSURF	4
FIGURA 2 ONDA INICIAL	6
FIGURA 3 CONJUNTO ONDAS	6
FIGURA 4 ONDAS FRECUENCIA	7
FIGURA 5 ONDAS DESCOMPUESTAS.....	7
FIGURA 6 ONDA CIRCULAR CORRECTA (NÚMERO ENTERO DE PERIODOS).....	9
FIGURA 7 ONDA CIRCULAR INCORRECTA (NÚMERO NO ENTERO DE PERIODOS)	9
FIGURA 8 FUNCIONES VENTANA.....	9
FIGURA 9 LOGOS PROYECTO.....	12
FIGURA 10 LOGOS CONTROLADOR_FOURIER	14
FIGURA 11 ENTRADAS Y SALIDAS CONTROLADOR_FOURIER.....	14
FIGURA 12 LOGOS ANALIZADORES.....	15
FIGURA 13 ENTRADAS Y SALIDAS ANALIZADORES	15
FIGURA 14 LOGOS CONTROLADOR	17
FIGURA 15 ENTRADAS Y SALIDAS CONTADOR	17
FIGURA 16 LOGOS CONTROLADOR_RESULTADOS	18
FIGURA 17 ENTRADAS Y SALIDAS CONTROLADOR_RESULTADOS	18

FIGURA 18 DIAGRAMA DE FLUJO	19
FIGURA 19 AMPLITUDES GUITARRA	21
FIGURA 20 DESARROLLO PROCESAMIENTO	22
FIGURA 21 LEONA LEWIS – I WILL BE	23
FIGURA 22 TAYLOR SWIFT - WE ARE NEVER EVER GETTING BACK TOGETHER.	23
FIGURA 23 ERROR RITMO	23
FIGURA 24 PROCESAMIENTO DE UNA SEÑAL	25
FIGURA 25 DESCOMPRIMIR AL CARGAR	26
FIGURA 26 CANALES INTERCALADOS	26
FIGURA 27 COMPROBACIÓN RITMO	28
FIGURA 28 COMPROBACIÓN RITMO 2	29
FIGURA 29 SONOGRAMA CANCIÓN TRISTE.....	31
FIGURA 30 CONTADOR UNITY	32
FIGURA 31 RESULTADO BPM	34
FIGURA 32 RESULTADO DEL RITMO	35
FIGURA 33 RESULTADO INTENSIDAD	35
FIGURA 34 RESULTADO FRECUENCIA	36
FIGURA 35 RESULTADO FINAL.....	36
FIGURA 36 RANKING CANCIONES	48
FIGURA 37 ESTUDIO RITMO	50
FIGURA 38 ESTUDIO CANCIÓN 2	51
FIGURA 39 ESTUDIO CANCION 7	52
FIGURA 40 ESTUDIO CANCIÓN 5	53
FIGURA 41 ESTUDIO CANCIÓN 9	54
FIGURA 42 COMPARACIÓN 1	54
FIGURA 43 COMPARACIÓN 2	55

FIGURA 44 COMPONENTES FRECUENCIALES	59
FIGURA 45 OSCILOGRAMA	59
FIGURA 46 OSCILOGRAMA. EJE VERTICAL(AMPLITUD), EJE HORIZONTAL (TIEMPO)	60
FIGURA 47 ESPECTROGRAMA.....	61
FIGURA 48 ONDA.....	63
FIGURA 49 ARMÓNICOS [REFERENCIA 7].....	64
FIGURA 50 AMPLITUD	65
FIGURA 51 OÍDO	66
FIGURA 52 CARACTERIZACIÓN TIMBRE.....	67

INDICE DE TABLAS

TABLA 1 LOGOS PROYECTO	13
TABLA 2 TENDENCIA DE FRECUENCIA.....	30
TABLA 3 CANCIONES ESTUDIO	41
TABLA 4 TABLA RESULTADOS	42
TABLA 5 GRAFICAS ANÁLISIS INDIVIDUALES.....	47
TABLA 6 BPM	49
TABLA 7 FOURIER CANCIONES TRISTES	56
TABLA 8 FOURIER CANCIONES ALEGRES	57

1. Introducción

En esta memoria se llevará a cabo la documentación del proceso de desarrollo de una herramienta que nos permita analizar los diferentes patrones de la música entrante, para poder utilizar esos datos en la generación de un entorno que responda al estado de ánimo generado por dicha música. En este apartado, se explican los diferentes motivos que se han tenido en cuenta a la hora de la realización del proyecto, así como los diferentes objetivos y la estructura a la que se ciñe el documento actual.

1.1 Motivación

Creado en un principio para distraernos, pero también para fomentar nuestra imaginación, los videojuegos forman cada vez más parte de la vida cotidiana de muchos jóvenes y de la sociedad actual en general. Su crecimiento y desarrollo en los últimos años ha ido ligado también a el impulso que se ha dado en la creación de juegos independientes, donde la importancia del apartado creativo como pilar en el diseño de una obra ha dado lugar a toda clase de desgloses en cuanto a temáticas para todos los gustos, gráficos, desarrollo en el apartado artístico y por supuesto en el apartado musical, que ha ido evolucionando paralelamente.

Por esta razón ha llegado un punto en el que no podemos concebir un juego sin su propia banda sonora que fluya a la par con la jugabilidad y el carácter artístico que se nos presenta.

La música en los videojuegos es un apartado creativo que ha evolucionado muchísimo en los últimos años. Las composiciones actuales han cambiado mucho respecto a aquellos sonidos de 8 bits que nos acompañaban antiguamente en las diferentes máquinas arcade, aunque la calidad de la banda sonora siempre ha sido un pilar importante a lo largo de la historia de los videojuegos, actualmente, contamos con composiciones mucho más ricas y de más complejidad musical que antiguamente. Esto es tan cierto, que incluso se pueden ver ya premios Grammys entregados a bandas sonoras de videojuegos, como paso en el 2011 con “Baba Yetu”, canción que aparece en el videojuego Civilization IV. Otros ejemplos de este impulso que se está llevando a cabo es la creación del proyecto “Final Fantasy: Distant Worlds”, que se trata de una orquesta que, mediante una gira mundial, transmite la música más reconocida de los juegos de Final Fantasy.

Esta época dorada de la música en los videojuegos tampoco queda oculta para los productores y artistas musicales, los cuales han podido ver el filón que supone el trabajar en la industria del videojuego, ya que además les aporta una gran libertad creativa a la hora de crear sus piezas, consiguiendo así composiciones de gran calidad musical.

Incluso hasta conocidas celebridades como Paul McCartney han aportado su grato de arena a este mundo de los videojuegos. Junto con O'Donnell y C. Paul Johnson, el antiguo Beatle, ha participado en la composición musical de Destiny. En este juego de género shooter podemos encontrar la canción “Hope for the Future” que encaja totalmente con la ambientación bélica de Destiny.

No solo en el extranjero se animan a participar en grandes obras musicales para un videojuego, en España tenemos a Óscar Araujo, que se animó a componer la banda sonora del juego Castlevania.

Podríamos seguir nombrando muchas más composiciones singulares que merecen un gran reconocimiento dentro del mundo de los videojuegos. Lo que no podemos negar es que, lejos del rechazo que causaba hace no tanto tiempo, el sector del videojuego está sufriendo un avance y desarrollo innegable en los últimos años, y su influencia e impacto se hacen notar más cada día.

En este Trabajo fin de Grado se ha querido aprovechar ese impulso y la gran importancia del apartado musical dentro de un videojuego, para desarrollar un código capaz de entrelazar más aun ambas cosas, consiguiendo así un resultado que responda a la sensibilidad musical y las emociones que las obras inspiran en el jugador. De esta manera se puede llegar a tener un entorno que cambie y se adapte según los sentimientos que evoca una determinada banda sonora.

1.2 Objetivos

El objetivo principal que tiene este proyecto, es el diseño e implementación de un sistema que nos permita llevar a cabo un análisis de distintos parámetros musicales pertenecientes a una determinada canción, con el fin de analizar el sentimiento que esta pretende influir en el jugador. De esta manera, y gracias a estos datos recogidos, se podrá crear un entorno dinámico para el jugador en el que vea como se producen diferentes cambios relacionados directamente con el apartado musical del videojuego.

Para alcanzar este objetivo final, se abordarán una serie de objetivos parciales que permitirán alcanzar los conocimientos necesarios para poder llegar al propósito final de forma progresiva. Estos objetivos parciales son los siguientes:

- Análisis de los parámetros más influyentes en la relación entre los sentimientos y la música.
- Estudio sobre la extracción de información musical en un audio y las propiedades más determinantes juntos con los valores de estas que se adecuan más al estudio que se va a realizar.
- Análisis sobre las técnicas más adecuadas para la extracción de los diferentes parámetros de una pieza musical.
- Desarrollo de una base de datos musical que contendrá diversas muestras de audio relacionadas con su respectiva información (tonalidad, ritmo, etc.). De esta forma se podrá comparar correctamente y de forma más eficiente los distintos prototipos que se desarrollen.
- Implementación de diferentes prototipos que detecten los parámetros considerados en el segundo punto.
- Comprobación de su eficiencia mediante el análisis los resultados obtenidos de forma experimental.
- Estudio de los resultados obtenidos frente a otro software existente. De esta forma se podrá comprobar la precisión que presentan a la hora de realizar el análisis de alguno de los parámetros.

- Describir de forma clara y precisa el funcionamiento de los algoritmos realizados, así como los diferentes pasos que hay que se deben realizar en la obtención de las mediciones correspondientes.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **1) Introducción:** Se explica el contexto que rodea a este proyecto, la motivación para su realización y los diferentes objetivos planteados.
- **2) Estado del arte:** Se comentan algunas de las tecnologías y herramientas existentes en relación con el proyecto.
- **3) Diseño:** Se describe el diseño del proyecto, explicando cada una de las clases con diferentes diagramas.
- **4) Desarrollo:** Se explica el proceso de desarrollo llevado a cabo.
- **5) Pruebas:** Se exponen las diferentes pruebas llevadas a cabo para la verificación del correcto desarrollo.
- **6) Conclusiones y Trabajo Futuro:** Se comentan algunas de las conclusiones a las que se ha llegado tras la realización del proyecto aquí expuesto. Por otra parte, se expondrán alguna serie de puntos de mejora de la solución planteada.
- **7) Referencias:** Fuentes de información en las que se basan algunas referencias expuestas en esta memoria.
- **8) Anexos:** Información adjunta y relevante para el diseño y realización de la solución planteada.

2 Estado del arte

En este capítulo de la memoria se hará un estudio sobre algunos videojuegos, herramientas o software y tecnologías que guardan relación con el tema que se presenta en este trabajo de fin de grado.

2.1 Videojuegos

La música es un tema fundamental en el desarrollo de un videojuego y es uno de los pilares más influyentes en el diseño de este. El uso de la música tiene una gran flexibilidad. Se puede encontrar la música en un sinnúmero de aspectos del juego, aunque lo más común suele ser encontrarla presente en la banda sonora, para poder proyectar una determinada atmósfera al jugador y conseguir así que este sienta una mayor inmersión en el juego, o bien en el desarrollo de las mecánicas del videojuego, creando y diseñando diferentes puzzles o comportamientos entorno a ella. A continuación, se mencionarán algunos de los videojuegos basados en este concepto más importantes de los últimos años y actuales.

2.1.1 SingStar

Se trata de un juego desarrollado por SCE London Studio y publicado por Sony que se lanzó en el 2007. En él, se puede competir contra otros jugadores cantando como si se tratase de un karaoke. El objetivo es ajustar el tono de tu voz lo más posible a el objetivo que se muestra en la pantalla para recibir más puntos. Se trata de un juego perfecto para fiestas con hasta 8 jugadores compitiendo entre ellos. Además, se tiene la posibilidad de usar tu teléfono móvil como micrófono y así no tener la necesidad de comprar uno.

2.1.2 AudioSurf

En AudioSurf se puede encontrar un juego basado totalmente en la música, donde el mapa varía según lo que esté sonando en ese momento. En este videojuego, el jugador es una especie de nave de carreras que tiene que recorrer un circuito con varios carriles, en los cuales se tendrá que ir recogiendo diferentes bloques de colores que se irán generando al ritmo de la música. Resulta muy interesante la posibilidad de incorporar una lista de música y ver como esta se transforma en una pista de carreras donde cada cuesta, cada giro y bloque encaja perfectamente con la canción que está sonando.



Figura 1 AudioSurf

2.1.3 Patapon

Este pequeño juego basado en el ritmo y cuyo nombre está formado por dos onomatopeyas japonesas (“para” para la marcha y “pon” para el sonido de tambores), fue lanzado al mercado en el 2007 y te pone en la piel de un dios que tiene que guiar a una pequeña tribu hacia la batalla contra un malvado reino. Todas las ordenes que le das a tu tribu se basan en apretar los botones correctos para así formar un ritmo con tus tambores sagrados, mediante los cuales, guiaras las decisiones de tu gente. Diferentes combinaciones de botones darán lugar a diferentes resultados de movimientos.

2.1.4 Guitar Hero 2

En este juego que vio la luz en el 2006 se emula una guitarra eléctrica mediante una serie de botones y palancas. Para jugar es necesario comprar un controlador que sé tendrá que conectar a la consola (en este caso una guitarra con una serie de botones). El jugador tendrá que apretar los botones de la guitarra en la secuencia correcta que se muestra en la pantalla, para conseguir la mayor cantidad de puntos. Este juego tuvo un mejor resultado en ventas que el Guitar Hero 1, ya que no solo aprovechaba todos los puntos fuertes de la primera entrega, si no que añadió una mayor personalización de los personajes, mejores controles y muchas más canciones.

2.2 Software para análisis de audio

Existen en el mercado multitud de sintetizadores y programas cuyo objetivo es el análisis del audio, así como otros programas como Matlab, que, sin estar orientados al análisis de audio como tal, tienen la posibilidad de realizar diferentes operaciones sobre señales de audio que resultan muy útiles, como por ejemplo la Transformada de Fourier. Esta sección estará dedicada a nombrar algunos de los más relevantes.

2.2.1 Audacity

Es uno de los editores de sonido más populares del mercado, ofrece una gran cantidad de efectos que puedes añadir a tus audios, así como un gran conjunto de herramientas como ecualizadores o analizadores de espectro. Es gratuito.

2.2.2 Adobe Audition CC

Conjunto de herramientas de carácter profesional pertenecientes a la marca Adobe. Te permite una visualización espectral de un audio, así como diferentes funciones de forma de onda, a la vez que una potente capacidad en la edición de sonidos.

2.2.3 SATLive

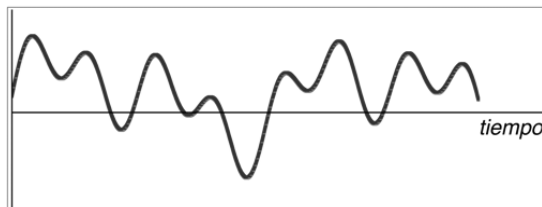
Se trata de un software de medición dual de audio basada en el FFT. Nos trae un gran abanico de herramientas preparadas para optimizar un sistema de sonido. Se utiliza principalmente en el sonido en vivo, pero es un gran programa a la hora de medición de audio en general.

2.2.4 Smaart

Una de las principales competencias de SATLive. Se trata de otra herramienta de análisis de audio más usadas en el mercado actual.

2.3 Transformada de Fourier

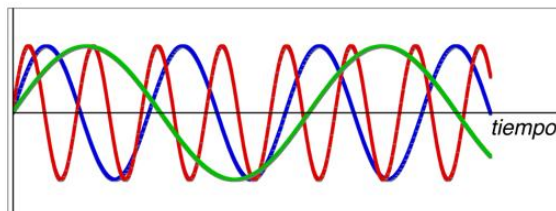
Al tratarse el sonido de una sucesión de frecuencias se hará uso de la transformada de Fourier para su correcto análisis y estudio. La transformada de Fourier se trata de un algoritmo matemático que permite descomponer un sonido en senos y cosenos (en ondas sinusoidales), con diferentes amplitudes y frecuencias. Gracias a esto se pueden apreciar las diferentes fases de los sonidos, su amplitud, su frecuencia, incluso diferenciar el tono y el timbre de la onda. El teorema de Fourier afirma que cualquier onda en el dominio del tiempo puede ser representada por la suma acumulativa de senos y cosenos. Esto se puede ver muy fácilmente en las siguientes imágenes:



Imaginemos que inicialmente tenemos esta onda.

Figura 2 Onda Inicial

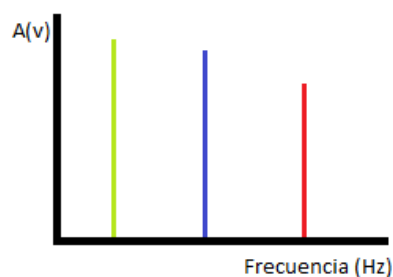
Según lo mencionado anteriormente, igual que uno puede formar ondas o señales mezclando o superponiendo una serie de senos y cosenos, también se puede analizar una onda en sentido inverso, y deconstruirla en senos y cosenos.



Como se puede apreciar, el conjunto de estas ondas, daría como resultado la onda de la imagen anterior.

Figura 3 Conjunto Ondas

Con la FT (transformada de Fourier) se va a realizar una conversión de una señal en el dominio del tiempo, a una señal en el dominio de la frecuencia.



En esta imagen se observan las 3 ondas anteriores en el dominio de la frecuencia.

Figura 4 Ondas Frecuencia

Un ejemplo en el que también se puede apreciar muy claramente es en las ondas cuadradas, formadas por un conjunto de armónicos:

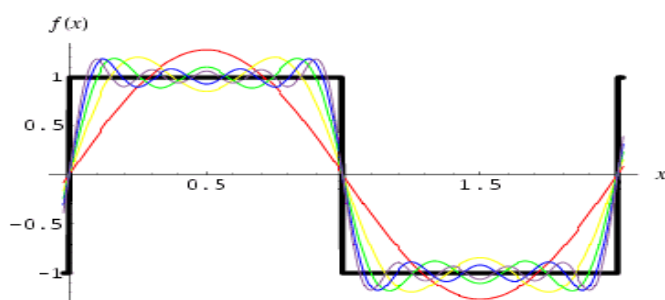


Figura 5 Ondas Descompuestas

Como vemos, esta señal se puede descomponer en un conjunto infinito de ondas, que sumándose entre ellas consiguen la peculiar forma cuadrada de la onda final.
[Referencia 11]

Comprendiendo que la transformada de Fourier nos permite transformar y deconstruir una señal en el dominio del tiempo a una señal en el dominio de frecuencia, hay que tener en consideración que el dominio de frecuencia tiene en cuenta las diferentes amplitudes que contienen las diferentes frecuencias. En general, simplemente se trata de una forma diferente de ver la misma señal original. Hay que saber también que un digitalizador va a muestrear una onda (su forma), y la va a transformar en valores discretos. Por esta razón, no se podrá usar la transformada de Fourier con estos datos. Sin embargo, si se puede utilizar la transformada de Fourier discreta, que se nombrará a partir de ahora como DFT. Esta se produce como resultado de que sus componentes de dominio de frecuencia, se encuentren en valores discretos.

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{-\left(2\pi n \frac{k}{N}\right)j} = \sum_{n=0}^{N-1} x[n] W_N^{nk},$$

$$k = 0, 1, \dots, N-1.$$

Transformada discreta de Fourier [Referencia 1]

- $x[k]$: Espectro medido del k ésimo elemento de la DFT, en función de la frecuencia discreta.

- N : Número de muestras que hay en $x[n]$. Se tienen en cuenta también los ceros.
- $X[n]$: Secuencia de datos discreta. N indica el índice de la muestra original.
- $e^{-j(2\pi n \frac{k}{N})} = W_N$: Factor de fase.

Aunque este tipo de transformada sí es posible implementarla en una computadora, tiene un costo computacional muy alto para secuencias largas. Necesitamos encontrar un modo de aplicarla con una mejor eficiencia.

Por otro lado, la transformada rápida de Fourier, a la que llamaremos FFT, se trata de una implementación más optimizada de la DFT que realiza los mismos cálculos. Con la FFT se realiza menos cómputo y su rendimiento es mejor basándose en la recursividad y aprovechando la simetría de W_N , a la vez que los errores de redondeo se hacen menores. [Referencia 2]

$$(W_N)^{nk} = e^{-j\frac{2\pi}{N}nk} = \cos\left(\frac{2\pi}{N}nk\right) - j * \sin\left(\frac{2\pi}{N}nk\right)$$

$$(W_N)^{n+N} = (W_N)^n$$

$$(W_N)^{n+\frac{N}{2}} = (-W_N)^n$$

Llegados a este punto en la explicación de la transformada de Fourier, se debe comprender que si bien, proporciona un gran análisis sobre una señal, también tiene ciertas limitaciones, como por ejemplo que solo puede aplicarse a un número de muestras que sea potencia de dos. Aun con estas limitaciones hay que tener en cuenta que si la DTF tenía un rendimiento aproximado de N^2 , la FFT alcanza aproximadamente un rendimiento de $N \log_n(N)$. [Referencia 3]

Cuando se usa la FFT para analizar una señal, el proceso de análisis se basa en la suposición de que se trata de un conjunto finito de datos, lo que quiere decir que es un segmento (un periodo) de una señal periódica. Para la FFT el dominio del tiempo y de la frecuencia se tratan de topologías circulares. Esto quiere decir que la onda tiene conectados sus extremos, de forma circular. Si se mide una señal y se trata de una onda periódica, con un número entero de periodos que llenan el intervalo de tiempo de adquisición, se produce una FFT correcta, ya que coincide con la suposición mencionada anteriormente. Por el contrario, hay que asumir que esto no siempre se cumple, y que es común que la señal medida no contenga un número entero de periodos. En estos casos se produce como resultado una forma de onda truncada, que difiere de las características de la señal de tiempo original, y se pueden producir cambios de transición brusca. Estas transiciones bruscas son discontinuidades, por lo que, si no se cumple que la señal original sea circular, la señal de salida resultará discontinua y errónea. Estas discontinuidades mencionadas anteriormente se representan en FFT resultante como componentes de frecuencias muy altas que no se encuentran en la señal original. En ese caso el espectro resultante no sería el espectro real de la señal original, si no una versión errónea. Este fenómeno se conoce como fuga espectral.

Para aclarar el concepto de señal circular, puede hacer uso de las siguientes imágenes:



Figura 6 Onda Circular Correcta (número entero de periodos)

Así, por ejemplo, como la siguiente onda no se trata de una onda circular, generaría una FFT errónea:

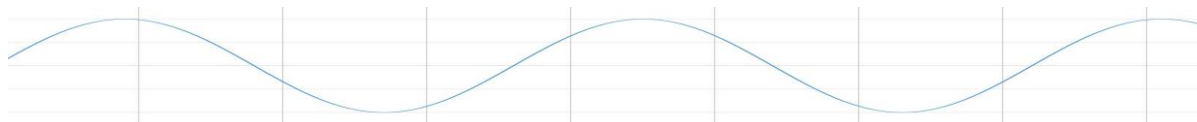


Figura 7 Onda Circular Incorrecta (número no entero de periodos)

La forma de eliminar estos defectos a la hora de aplicar la FFT con una onda no circular, es el uso de una técnica llamada “funciones ventana”. Estas funciones se encargan de reducir la amplitud de las discontinuidades de la onda en los límites de nuestro intervalo analizado. El procedimiento consiste en multiplicar el intervalo de tiempo, por una ventana de longitud finita, cuya amplitud, tenderá poco a poco hacia cero en los bordes. De esta manera, ambos extremos de nuestra señal se encontrarían, y tendríamos como resultado una onda circular sin ningún tipo de transición brusca, y por lo tanto, sin fuga espectral. Esto se conoce como “*aplicar una ventana*” al FFT.

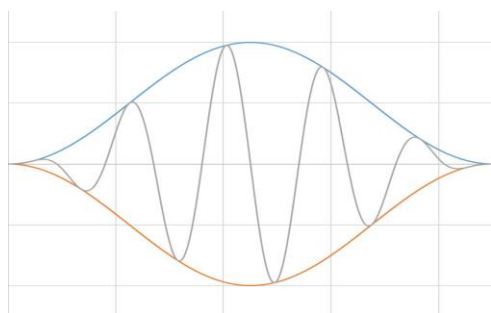


Figura 8 Funciones Ventana

Uso de funciones ventana para minimizar el efecto de la fuga espectral.

2.4 DSPLib

Mediante el uso de esta librería en el desarrollo de nuestro algoritmo, podremos realizar la Transformada rápida de Fourier o la Transformada discreta de Fourier en C# para poder obtener así los datos de espectro necesarios para su posterior análisis. Esta librería cuenta también con un gran conjunto de ventanas que se pueden aplicar a la Transformada realizada. En conceptos generales, esta librería nos va a permitir aplicar la FFT a un

conjunto de datos que se le proporcionarán. Se hará uso de esta librería en el preprocesado de un archivo de audio, que se podrá ver más adelante en la sección de desarrollo.

Aunque gracias a DSPLib, se tiene acceso a varios tipos de análisis y de Transformadas de Fourier, en este trabajo se limitará su uso a la FFT (Transformada rápida de Fourier), ya que se trata de una transformada mucho más rápida que la DFT (Transformada discreta de Fourier). Su uso es relativamente sencillo, y en su documentación vienen multitud de ejemplos para poder realizar un primer acercamiento sin muchas dificultades. Para su utilización es necesario añadir un tipo de datos que no viene implementado en Unity, por lo que habría que añadirlo manualmente de la librería System.Numeric de C.NET, y realizarle unos pequeños cambios para su correcto funcionamiento.

2.5 Psicología de la música

Tratándose gran parte de este proyecto sobre la influencia de la música en los sentimientos humanos, estaríamos cometiendo un gran error si no comentásemos nada sobre la psicología de la música, la cual se encarga precisamente de esto. Hoy en día es evidente que la música nos afecta directamente. Se trata de una parte esencial en nuestra vida diaria, y tiene cierto influjo en cómo se comporta nuestro cerebro, órgano susceptible a la adaptación.

En esencia, la música no es más que sonido, vibraciones y energía. Esta llega a nuestro oído y del oído a nuestro cerebro, activando así ciertas zonas de este. Esta más que demostrada la relación entre estas vibraciones y el efecto que nos causan, teniendo influencia directa en nuestra vida emocional y afectiva.

Entendemos la psicología de la música como el estudio de la fuerza emocional que puede llegar a ejercer la música en el estado de ánimo de las personas, teniendo una influencia directa en su comportamiento. Esta fuerza es aplicable en cualquier etapa de la vida de una persona.

Existen multitud de investigaciones llevándose a cabo en este campo, y cada día más centros están interesados en el desarrollo de un correcto método de enseñanza para esta rama de la psicología, la cual exige grandes conocimientos en psicología y un gran dominio sobre música. [Referencia 20]. Y es que, la música no solo es capaz de producir un cambio cognitivo en los seres humanos, influyendo en como procesan la información, comprensión, capacidad de retención o de atención. Si no que también es capaz de producir un cambio físico del cerebro mediante la enseñanza y práctica de la música. Desarrollándose ciertas partes de este. Por ejemplo, la zona encargada de registrar y diferenciar los estímulos acústicos es de media un 25% más grande en personas que han tocado algún instrumento. [Referencia 21]

3 Diseño

En esta sección se va a proceder a describir todos los detalles que han influido a la hora de llevar a cabo la implementación de nuestro proyecto. De forma previa al comienzo del diseño de la solución planteada, se lleva a cabo un proceso de investigación en el que se aclaran diferentes conceptos:

- Análisis del concepto de música.
- Análisis del funcionamiento del sonido.
- Estudio de las características de frecuencia y amplitud musical.
- Relación entre el sonido y como lo percibimos en el oído humano.
- Comprensión de las características del sonido más influyentes en el estado de ánimo de las personas.

Posteriormente a esta investigación se realiza un estudio con una serie de participantes. El objetivo de este estudio es la comprobación de forma experimental de los conceptos adquiridos hasta el momento. Gracias a él, se nos permite comprobar cómo influyen las diferentes características de una pieza musical de forma individual en cada uno de los participantes. De esta forma, se tiene un mayor conocimiento a la hora de poder valorar los límites de las diferentes características que analizaremos, frente a los sentimientos que emanan en el oyente/jugador y poder realizar una ordenación de estas respecto a el estado de ánimo que generan. Esta información puede encontrarse en :

- Anexo 8.5 “Conocimientos previos al diseño”
- Anexo 8.3 “Estudio sobre la música”

A continuación se procederá a listar los diferentes **requisitos** del proyecto presente los cuales definirán las propiedades del proyecto realizado en este TFG:

R.1: Capacidad de procesamiento de las características sonoras de una determinada pieza musical.

R.2: Análisis correcto de diferentes canciones independientemente del género musical al que pertenezcan.

R.3: Las diferentes características de una canción deberán analizarse de forma individual.

R.4: Posibilidad de configurar de la ponderación cada uno de los algoritmos de análisis implementados. Cada una de las características analizadas vendrá acompañada de un multiplicador (ponderación) que indicará su peso en el resultado final del sentimiento generado por una canción.





R.5: Capacidad de adicción o sustracción de algoritmos que se encarguen del análisis de nuevas características.

R.6: Análisis de sentimientos básicos como son: Alegría, tristeza e ira.

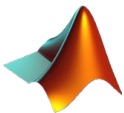


En este proyecto se ha hecho uso de diferentes tecnologías tanto para la parte de programación e implementación de una solución valida, como para un correcto control sobre las versiones implementadas, y por supuesto, para un correcto estudio del diseño de la solución propuesta. A continuación se muestran algunas de las principales tecnologías utilizadas en la totalidad del proyecto:



Figura 9 Logos Proyecto

	El desarrollo de este proyecto ha sido realizado en el motor de videojuegos multiplataforma Unity .
	Se ha hecho uso del lenguaje de programación C# para la implementación de los diferentes algoritmos.
	Como entorno de desarrollo se ha utilizado el Microsoft Visual Studio . En él se han programado los diferentes algoritmos y se han llevado a cabo las diferentes depuraciones.
	Se hace uso de la librería DSPLib para la implementación y uso de la transformada de Fourier y la obtención de los diferentes datos para su posterior análisis.

Para la elaboración de diferentes pruebas y comprobaciones previas al desarrollo de los algoritmos finales se hace uso de varios programas de análisis de audio. Esto permite estudiar las características musicales más influyentes de una pieza musical, como se comporta, su forma de onda, amplitudes, frecuencias y el estudio de su transformada de Fourier (en cualquiera de sus variantes). Los 3 software principalmente utilizados se muestran a continuación:

	Matlab permite la implementación de la transformada de Fourier y la obtención de diferentes gráficos para el estudio de las características de una canción.
	WavePad permite analizar el oscilograma de un sonido y hacer uso tanto de su FFT como de su TFFT.
	Por último Audacity permite también realizar un estudio del oscilograma de una canción, así como de realizar un análisis de espectro.

Por último, para el control de versiones se ha utilizado:


	Se hará uso de un repositorio en GitHub , donde se irán almacenando todos los diferentes cambios que se efectúen en el proyecto. Estos cambios se almacenarán mediante commits tras cualquier tipo de modificación.
---	--

Tabla 1 Logos Proyecto

Uno de los objetivos principales que se han buscado durante la realización de este proyecto es la capacidad para escalarlo y modificarlo de manera sencilla. Se pretende que el proyecto pueda ser ampliado y modificado dependiendo del objetivo que se busque, o de los diferentes resultados que uno quiera obtener de él. Por esta razón se intenta que la estructura que se va a definir a continuación sea lo más escalable posible, y se encuentre correctamente encapsulada.

El objetivo global del proyecto es poder realizar el análisis de una serie de características musicales como son el ritmo, frecuencias máximas etc. con el fin de poder obtener las diferentes salidas de cada uno de estos algoritmos y poder generar un resultado final que represente la emoción generada por la canción que se esté reproduciendo actualmente.

A continuación se describirán los diferentes componentes en los que se ha dividido el proyecto:

3.1 Controlador_fourier

En esta clase se llevará a cabo la obtención de los diferentes datos de la Transformada rápida de Fourier gracias a la librería DSPLib. A medida que se obtienen los datos de espectro, se utilizarán para analizar las diferentes características musicales de la pieza sonora que se está reproduciendo.

Las **tecnologías** utilizadas en esta clase son las siguientes:



Figura 10 Logos Controlador_fourier

- Se hace uso tanto del programa visual estudio como del motor de videojuegos Unity para la elaboración del código en C#.
- Se utiliza la librería DSPLib para el cálculo de la FFT.

Las **entradas y salidas** son las siguientes:

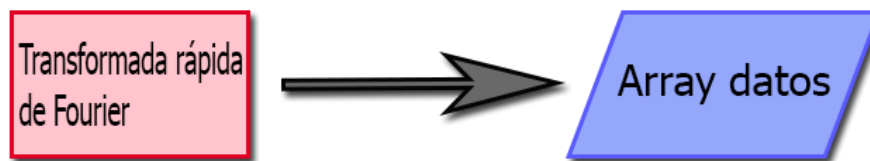


Figura 11 Entradas y Salidas Controlador_fourier

- Al comenzar la reproducción del juego, este se encargará de comenzar el cálculo de la FFT, haciendo uso de la librería DSPLib.
- Este Array de datos de la FFT será usado por el conjunto de algoritmos encargados de analizar las características del audio que se está reproduciendo.

3.2 Analizadores

Los analizadores serán un conjunto de clases que se encargarán de los diferentes estudios sobre las características musicales de la canción en reproducción. Cada uno de los analizadores existentes tendrá como objetivo, únicamente, una de estas características, y contendrán una variable numérica que indicará la ponderación o peso que tendrá esa característica en el resultado final del sentimiento generado por cada canción. Harán uso de la información proporcionada por el ControladorFourier.

Las **tecnologías** utilizadas para el diseño de estas clases son las siguientes:



Figura 12 Logos Analizadores

- Se hace uso tanto del programa visual estudio como del motor de videojuegos Unity para la elaboración del código en C#.
- Se utilizan diversos programas de análisis de audio para la comprensión de las características musicales. Este estudio nos permitirá un diseño correcto de cada uno de los analizadores, así como la generación de pruebas y la comprobación de los resultados obtenidos.

Las **entradas y salidas** son las siguientes:



Figura 13 Entradas y Salidas Analizadores

- Se hará uso del Array de datos generado por el script ControladorFourier. Estos datos contendrán los datos de espectro resultado de la FFT.
- Si hay suficientes datos se realizará el análisis. Cada uno de los scripts de análisis se encarga del estudio de una de una característica concreta de la música.
- Se generará como resultado un número del 0-1 que nos indicará lo triste o alegre de los datos de la característica analizada. Por ejemplo, si se trata del analizador del ritmo, y el ritmo analizado es bajo (propio de una canción triste), devolverá un valor cercano a 0.

Hay que tener en cuenta cada uno de los scripts no analizará la totalidad de una canción y darán un resultado general a todo lo analizado. Las canciones, por norma general, no se comportan igual durante toda la pieza musical. Es común la existencia de cambios de ritmo, momentos con más intensidad, estribillos, etc. Si no se analizase la canción por intervalos, dividiéndola en diferentes lapsos de tiempo y generando un valor “sentimental” (concepto que se refiere a la emoción generada) diferente para cada uno de estos

fragmentos, se estaría ignorando las diferencias en cada uno de los intervalos de una canción.

Los **analizadores implementados** son los siguientes:

3.2.1 Ritmo

Este analizador se encargará de detectar los BPM de una canción (beats per minute). Se trata de una de las características principales que se pueden estudiar en una de las canciones que se reproduzcan. Ritmos lentos son característicos de canciones más tristes que los ritmos rápidos, los cuales suelen aparecer en canciones más alegres.

3.2.2 Tendencia de Frecuencia

Una de las características más importantes a la hora de calificar la emoción generada por una canción, y estudiada en el anexo 8.3-Estudio sobre la música. Consiste en analizar la canción en reproducción para ver que frecuencias son las dominantes.

Al tratarse de una canción triste, suelen aparecer frecuencias más bajas. Por otra parte, las canciones alegres, suelen caracterizarse por frecuencias más elevadas.

3.2.3 Silencios

Las canciones tristes suelen tener un número reducido de instrumentos. En la mayoría de las canciones analizadas es característico el encontrar 1 o 2 instrumentos acompañados de un vocalista. Tanto los instrumentos como el vocalista interpretan la pieza con un nivel de intensidad bajo, muy calmado. Por esta razón son muy características las intensidades que podemos apreciar en este tipo de canciones, donde las diferencias entre las intensidades de las frecuencias presentes en la canción, suelen ser muy reducidas, obteniendo así un abanico de valores muy pequeño, ya que las diferencias entre las máximas y las mínimas suelen ser escasas.

Por otro lado, las canciones alegres suelen tener un abanico de intensidades mucho más amplio, ya que la media de la canción suele ser mucho mayor que cualquier momento de esta en la que no estén sonando instrumentos o vocalistas.

3.2.4 Tendencia de intensidad

En este apartado se analizará la intensidad media de la canción. Las canciones tristes se suelen caracterizar por un nivel de intensidad medio más reducido que las canciones alegres.

3.3 Contador

De media, un juego con nivel de FTS considerado como aceptable, se ejecutará al menos a 60FPS. Eso significará que si se calculan los diferentes resultados aproximadamente cada 0,016 s. Una frecuencia que puede resultar excesiva llegando a afectar al rendimiento de nuestro juego.

Esta clase se encarga de controlar la frecuencia en la que los diferentes resultados son solicitados, evitando así realizar un simple Update() en nuestro script, de forma constante. Se indicará cada cuanto se debe actualizar el **resultado de la canción** (*), informando así a los debidos scripts encargados de facilitarlo, y relacionando el tiempo actual de reproducción de la canción con la posición del Array de datos de espectro.

***Resultado de la canción:** Valoración general del sentimiento que emite cada una de las canciones. Viene representada por un número.

Las **tecnologías** utilizadas para el diseño de la clase contador son las siguientes:



Figura 14 Logos Controlador

- El script está programado en C# mediante el programa visual estudio.
- Se hace uso del script en un GameObject del motor de videojuegos Unity.

Las **entradas y salidas** son las siguientes:

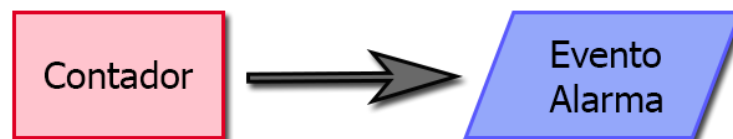


Figura 15 Entradas y Salidas Contador

- La clase contador no tendrá ningún tipo de entrada. Realizará una cuenta atrás según el tiempo que le indiquemos de frecuencia entre cada toma de resultados.
- Generará un evento de Unity que funcionará de alarma o TimeOut para los scripts subscriptos a él, indicando que tienen que calcular los resultados.

3.4 Controlador_resultados

Clase encargada de la gestión de los diferentes resultados. Cada uno de los valores devueltos por los diferentes analizadores deberán ser analizados dependiendo de la ponderación de cada uno. De esta forma, se podrá generar un resultado final que indicará el sentimiento que posee la canción que se está reproduciendo en ese momento. El resultado será utilizado para gestionar la intensidad de los diferentes efectos(*) implementados.

*Efectos: Diferentes respuestas implementadas para reaccionar al sentimiento de la canción en reproducción. Por ejemplo, luminosidad, niebla, rayos ... etc.

Las **tecnologías** utilizadas para el diseño del controlador de resultados son las siguientes:



Figura 16 Logos Controlador_resultados

- El script está programado en C# mediante el programa visual estudio. Este script se encargará de gestionar las operaciones relacionadas con el cálculo del resultado final de la canción.
- Se hace uso del script en un GameObject del motor de videojuegos Unity.

Las **entradas y salidas** son las siguientes:

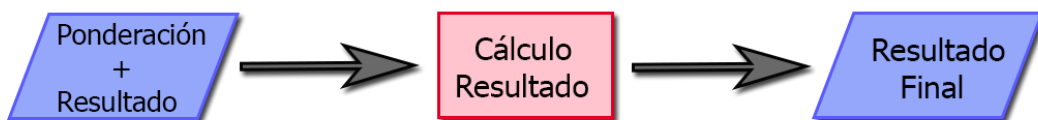


Figura 17 Entradas y Salidas Controlador_resultados

- Controlador_resultados usará los valores de los diferentes scripts de análisis. Estos scripts suministrarán no solo el valor de la característica que hayan analizado, sino también el valor de ponderación. Esta ponderación nos indicará el peso que tiene esa característica en el resultado final.
- El script Controlador_resultados realizará los cálculos necesarios para obtener el resultado final del del sentimiento final de la canción, para ello hará uso de los valores y ponderaciones mencionadas anteriormente.
- El resultado final será utilizado por los diferentes efectos en la escena, para su propia configuración e intensidad.

3.5 Diagrama de flujo

El **diagrama de flujo** del proyecto presente, es el siguiente:

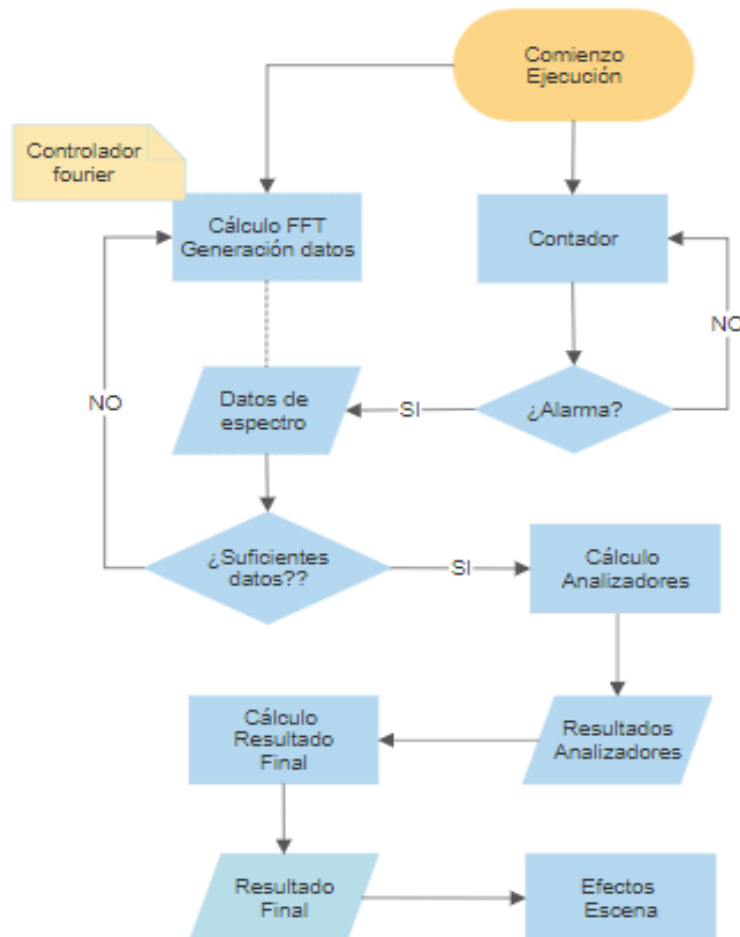


Figura 18 Diagrama de flujo

1. Comienza la ejecución.
2. Comienza la cuenta regresiva de la Clase Contador, y también se comienza a calcular la FFT, almacenando los datos de espectro en un array.
3. Cuando el contador genera el evento alarma se intenta realizar el cálculo de los diferentes resultados en cada analizador. Los analizadores deben comprobar que tienen los suficientes datos de espectro para poder analizar correctamente cada característica.
4. Cada uno de los diferentes analizadores genera un resultado parcial, propio de la característica que tengan como objetivo analizar.
5. Mediante todos los resultados parciales y sus ponderaciones, podremos llevar a cabo la obtención del resultado final de la canción actual.
6. Los efectos en la escena se actualizarán en base al resultado final obtenido anteriormente.

4 Desarrollo

Tras realizar toda la investigación con el fin de realizar el algoritmo propuesto, en este apartado del proyecto, se procederá a explicar la implementación de los diferentes prototipos y soluciones que se han desarrollado.

4.1 *Controlador_fourier*

Para poder utilizar y reproducir audios en Unity, se hace uso de un componente llamado AudioSource. Este componente permitirá reproducir cualquier fichero de audio, llamados AudioClip. En el presente proyecto se ha añadido este componente al GameObject Main Camera, con la propiedad “Play On Awake” activada para que empiece a reproducir la pista de audio cuando comience a ejecutarse. Unity nos aporta algunas herramientas muy útiles para poder acceder a la información del audio que se está actualmente reproduciendo. Una de estas herramientas es la función AudioSource.GetSpectrumData, la cual vamos a comentar a continuación

GetSpectrumData permite obtener una matriz que representa la amplitud relativa del audio analizado, en el dominio de la frecuencia. El resultado es solo de un intervalo del tiempo total del audio, y únicamente de un canal en particular, por lo que se tendrá que tener en cuenta si el audio está en estéreo. Lo que realmente está realizando la función GetSpectrumData es la Transformada Rápida de Fourier (FFT) para así poder transformar la amplitud del dominio del tiempo a el dominio de la frecuencia. Generalmente, y como se ha realizado en este proyecto, al realizar la FFT se aplica en todo el rango de frecuencias, que iría desde 0Hz hasta la mitad de la frecuencia de muestreo, aproximadamente un valor entre 44,1kHz – 48kHz (ya que la segunda mitad de la frecuencia de muestreo es una copia de la primera mitad de forma invertida, como si fuese un espejo). Por lo que nos quedaría una FFT entre 0Hz-24kHz. Esto quiere decir que la cantidad de muestras de audio que necesitaremos para realizar nuestra FFT será el doble del número devuelto por la función GetSpectrumData (ya que se necesita generar la segunda mitad), lo que puede crear cierta confusión a la hora de saber en qué momento se ha detectado cierta frecuencia.

En este punto del desarrollo se genera un problema, ya que a mayor del nivel de detalle que se necesite en las frecuencias analizadas, mayor será la necesidad de tiempo y muestras de audio que exigirá la función para generar ese nivel de detalle. Esto quiere decir que, si se quiere x rangos de frecuencias, se necesitarán $x*2$ muestras de audio. Se ha considerado suficiente 1024 rangos de frecuencias para el propósito de este proyecto, lo que aporta un nivel de detalle de unos 23.43Hz en cada muestra (teniendo en cuenta que la frecuencia de muestreo sea 48kHz, entonces $24\text{kHz}/1024 = 23.43\text{Hz}$) y exige 2048 muestras de audio.

Para saber la frecuencia de muestreo del Clip de audio se está reproduciendo en cada momento, basta con ejecutar la función AudioSource.clip.frequency. De esta manera se podrá calcular concretamente el número de frecuencias presente en cada una de las muestras.

La función `GetSpectrumData` tiene el siguiente aspecto:

`GetSpectrumData(float[] muestras, int canal, FFTWindow window):`

- **Muestras:** Número de muestras de frecuencias mencionadas anteriormente (1024). Esta función poblará este array con las muestras conseguidas en cada instante y debe tener un tamaño mínimo de 64 y máximo de 8192 y debe ser obligatoriamente, potencia de 2.
- **Canal:** Indica el canal de audio con el que se van a obtener los datos. Al tratarse de estéreo existen dos canales, pero existe la opción de indicar como canal el canal 0, el cual la media de ambos canales de estéreo combinados en un solo canal mono.
- **Window:** Función ventana que se aplicará a las muestras. Se puede elegir entre varias ventanas para realizar la FFT. Utilizaremos la ventana BlackmanHarris que reduce las fluctuaciones laterales en el dominio de la frecuencia sin perder mucha información en los límites .

4.2 Analizador: Ritmo

En la siguiente implementación del algoritmo de ritmo, se establece como objetivo el análisis de la onda con el fin de detectar un patrón en ella. Un aumento de intensidad que se repita de forma constante. Si se analizan las ondas sonoras emitidas por las notas tocadas en diferentes instrumentos, se puede obtener cierto patrón. Por ejemplo, he aquí la muestra de las amplitudes de una nota tocada en una guitarra.

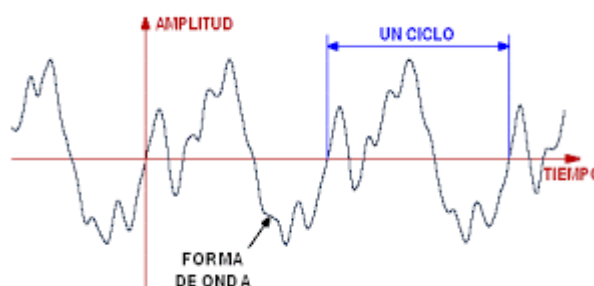
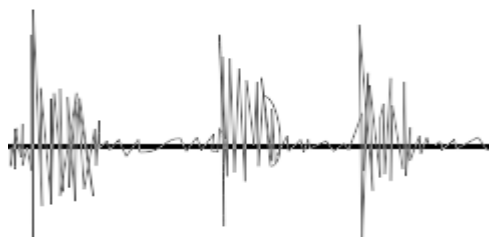


Figura 19 Amplitudes Guitarra

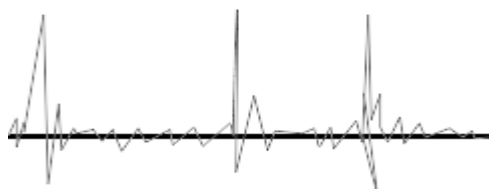
Se puede apreciar que el comienzo del ciclo consiste en una primera elevación o inicio, seguida por otra subida más drástica que llega hasta el límite de la amplitud de la onda, para después descender gradualmente. [Referencia 23]

Para detectar este patrón, lo primero que se lleva a cabo es la obtención de los datos del audio. Para esta tarea se hará uso de la función `GetSpectrumData` la cual estaremos ejecutando en nuestro script `Controlador_fourier`. Este script mediante la función nos aportará la información analizada mediante FFT en una muestra de audio concreta. Con la información obtenida de esta función, el siguiente objetivo va a ser el tratamiento de esta señal para la corrección y eliminación de la parte “estática” de la onda. Por último, se tendrá que localizar los patrones que quedan en la onda final. [Referencia 24].

Una imagen tosca de lo que queremos llevar a cabo, sería la siguiente:



Onda sin ningún tipo de procesamiento.



Onda procesada.



Localización del patrón.

Figura 20 Desarrollo Procesamiento

Para poder eliminar lo que se ha llamado “parte estática” de la información obtenida de `GetSpectrumData()`, se tendría que comparar varios fragmentos contiguos de la canción con el objetivo de analizar las similitudes que existen entre ellos y eliminar ruido. El objetivo sería reducir esas frecuencias presentes y comunes, para así obtener la parte que restante y que difiere del resto. De esta forma el patrón quedaría mucho más “visible”. Para poder hacer esta comparación, se va dividiendo los datos del Array de espectro y se almacena la última medida de frecuencias tomadas con el propósito de poder comparar con la del fragmento actual. Esto quiere decir que cada vez que se analicen los valores en el array pasado a `GetSpectrumData()` y que nos aporta `Controlador_fourier`, se tendrá que guardar una copia de los valores de cada división, para poder comparar con el valor de la división que está a continuación. A este concepto lo llamaremos diferencia espectral.

Para poder hacer esta comparación, primeramente se obtendrán las muestras del array de espectro. Estas muestras habrá que copiarlas a otro `Array[]` al que se ha nombrado como `antiguoEspectro`. Esto permitirá tener las muestras de la división anterior, para así poder comparar con ellas cuando se esté en la división siguiente. Una vez que se esté al menos en la segunda división analizada, se podrá comparar cada uno de los grupos de frecuencias en cada índice, con los grupos de frecuencia en los mismos índices pero de los datos obtenidos de la división anterior. De esta forma, se podrá restar cada uno de los índices de la división anterior, con los nuevos, y el resultado se guardará en un tercer `Array[]` donde se tendrá la “diferencia de espectro”, y será el espectro del frame actual. Este valor tiene que volverse a guardar en el `Array[]` de `antiguoEspectro`, ya que servirá para hacer la comparación con las medidas de frecuencia de la siguiente división. Al realizar la diferencia entre ambos espectros disponibles, solo se guardarán los números positivos, de esta forma se podrá saber si nos encontramos en un frame con un incremento espectral. En Unity nos quedaría algo así:

Para los índices entre $i=0$ y el $i=1024$

```
diferenciaEspectro[i] = Mathf.Max(0f, actualEspectro[i] - antiguoEspectro [i]);
```

Para poder conocer que tan drástico tiene que ser el aumento en la diferencia de espectro entre una división y otra para considerarlo un ritmo se tienen que tener en cuenta las fluctuaciones anteriores a lo largo de la canción, y de esta forma poder generar un intervalo, o un umbral, en el que, aunque las frecuencias aumenten, no se considere un ritmo. Esta medida de la variación mínima del espectro tendrá que tener en cuenta los cambios de la pieza musical alrededor de la división reproduciéndose actualmente.

Con esta solución en mente, se analiza una serie de canciones para valorar la viabilidad de las medidas que se han propuesto. Para la gran mayoría de las canciones no generaría ningún tipo de problema, pero sin embargo se puede encontrar estructuras que cambian drásticamente, ya sea gradualmente o por ejemplo al llegar a un estribillo. A continuación se muestran algunos ejemplos:

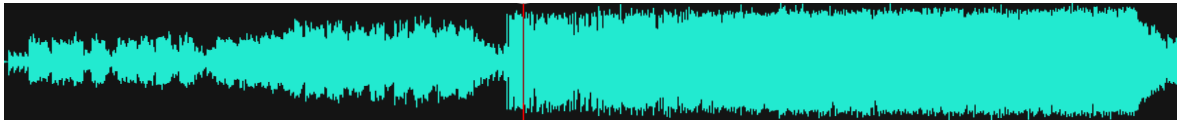


Figura 21 Leona Lewis – I will be.



Figura 22 Taylor Swift - We Are Never Ever Getting Back Together.

En todas estas canciones se puede ver claramente como, después de un intervalo de la canción donde las frecuencias tienen una determinada intensidad, se produce otro totalmente diferente. Este segundo intervalo contiene unas intensidades más elevadas, que afectarían negativamente a la detección del patrón. Nuestra solución de llevar una media de las fluctuaciones pasadas no soluciona este problema, ya que aunque la fluctuación del espectro que está sucediendo actualmente sea mucho mayor que la media de las fluctuaciones transcurridas hasta el momento, no implica que se trate del ritmo.

Por ejemplo, si se amplía una de las secciones de la canción I Will be de Leona Lewis mostrada anteriormente, se puede observar apreciar lo siguiente:

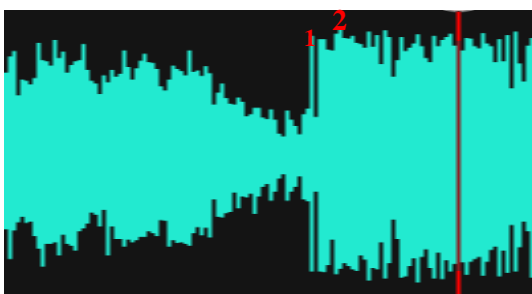


Figura 23 Error Ritmo

Aunque la solución propuesta marcaría el punto 1 como el ritmo, ya que se produce una gran subida en la intensidad de las frecuencias, es el punto 2 el que realmente es el ritmo de la canción.

Esto genera un nuevo problema, ya que no solo habría que tener en cuenta las fluctuaciones pasadas en las medidas de frecuencia, si no que habría que tener en cuenta también las futuras para anticiparse a posibles cambios en la intensidad media.

Unity no permite adelantarse y ver el flujo espectral de las muestras de frecuencia futuras de forma sencilla, por lo que la única opción que tendríamos sería aplicar un delay en la toma de muestras. Esto implicaría que el efecto que se aplicase iría totalmente desacompañado a el cambio que lo ha desencadenado. Por esta razón, es necesario buscar algún método que permita preprocesar la señal saliente del componente Audio Source, para poder anticiparnos a los cambios en las diferentes canciones. Estos cambios se realizarán en la clase `Controlador_fourier`, ya que como se verá a continuación, comenzará a hacer uso de threads y de la librería `DSPLib`. De esta forma podríamos tener acceso al array de datos preprocesado. Para la finalización de este analizador de ritmo, supongamos que podemos acceder a las muestras futuras. De esta forma se podrá terminar de exponer el funcionamiento de este analizador, para explicar en la sección siguiente como preprocesar el audio.

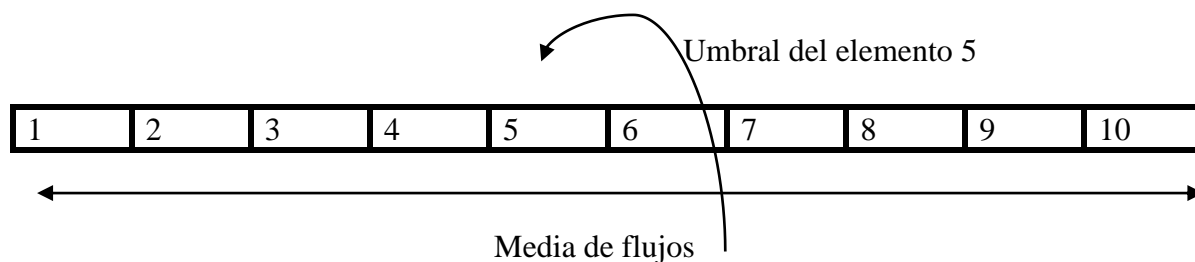
Se recuerda que calcular nuestro umbral va a servir para ver que frecuencias se están repitiendo en las muestras anteriores, y así poder eliminarlas de la actual. De esta forma es posible eliminar todo el ruido y obtener únicamente los cambios producidos en este instante. Para ello, se escogerá un tamaño de muestras para hacer el umbral adecuado con el que se considere que se tienen muestras suficientes de flujos de espectro y poder obtener una media de frecuencias presentes. Se sumarán todas esas medidas de flujos de frecuencias seleccionadas, y se realizará la media. Con este dato se recortarán los valores de la muestra tomada en el instante actual.

Una vez calculado el umbral de cada muestra, será multiplicado por una variable llamada “sensibilidadUmbral”. Como es evidente, esta variable permitirá manejar cuan sensible es el umbral del algoritmo, permitiendo regularlo como se desee.

Una vez finalizados los cálculos para saber que umbral se tiene en nuestra muestra actual, siendo el umbral la media de las muestras de flujo anteriores, se va a proceder a restárselo a el flujo de la muestra actual. De esta forma se podrá eliminar todas esas frecuencias comunes, como se puede apreciar en la primera imagen del desarrollo de este segundo algoritmo.

Por si no queda del todo claro, se adjunta un pequeño ejemplo. Imagine que tiene un array de 10 posiciones. En este array se han guardado los datos de los flujos espectrales de 10 muestras diferentes. Supongamos entonces que el tamaño del array de umbrales deseado para poder hacer la media es también 10. Esto significa, que al tener 10 muestras del flujo de espectro diferente, se puede calcular el umbral para un elemento. En este caso, el elemento 5, ya que es el elemento central (en este caso al ser par, queda descompensado) que tiene muestras por arriba y por abajo.

Lo que se haría entonces, es sumar el flujo de espectro de los diferentes elementos del 1 al 10 y posteriormente se calcularía su media. Este sería el umbral del elemento 5.



Si al restárselo a nuestro flujo actual, su umbral da un número menor de 0, significará que en esa frecuencia hay una media menor de la esperada. En ese caso será descartada y se quedará simplemente con 0, ya que no puede tratarse del ritmo (al ser menor). A el resultado lo llamaremos flujo espectral recortado.

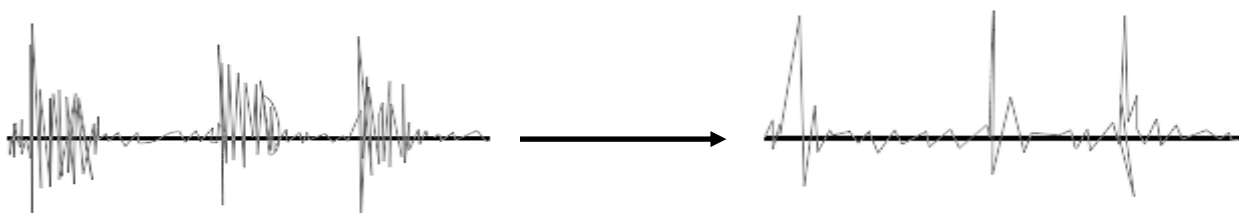


Figura 24 Procesamiento de una señal

Hay que tener en cuenta que, como se ha explicado anteriormente, el flujo espectral que se va a valorar no se trata del flujo espectral de las últimas muestras analizadas, ya que podría darse la situación que a partir de ese momento la media de intensidad en esas frecuencias, fuese mayor. Por esta razón se compara el penúltimo flujo espectral recortado analizado, de tal forma que solo será un pico en el ritmo, si es mayor, tanto del flujo espectral recortado de la muestra anterior, como del flujo espectral recortado de la muestra siguiente. Siguiendo el hilo del ejemplo anterior, al haber calculado el flujo espectral recortado del elemento 5, se podría comparar el flujo espectral recortado del elemento 4, con el del elemento 3 y 5. De esta manera se podría ver si supera las frecuencias posteriores y anteriores.

4.3 Preprocesamiento de señales de audio – Controlador_fourier

Aun con las facilidades que nos aporta Unity para poder analizar y aplicar la FFT a un sonido que está siendo reproducido, no resulta tan sencillo a la hora de llevar a cabo el mismo procedimiento para pre-analizar un sonido que, aunque definido, no está siendo reproducido aún. Aun así, hay ciertas herramientas con las que se puede llevar a cabo este objetivo.

AudioClip.GetData -> public bool GetData(float[] data, int offsetSamples)

Como indican en la propia documentación de Unity, esta función permite acceder a toda la información almacenada en un clip. Al igual que en `GetSpectrumData()`, el primer parámetro es el Array que la función se encargará de poblar con el data del audio. Por otra parte, el segundo parámetro, indicará a `GetData()` desde donde tiene que empezar a leer el audio. En este proyecto será 0, ya que se quiere toda la información desde el comienzo de la pista.

Para aplicar esta función a un audio, es esencial que al importar el clip, este marcado como “Descomprimir al cargar”, si no, el Array[] Data, simplemente será rellenado de 0’s.

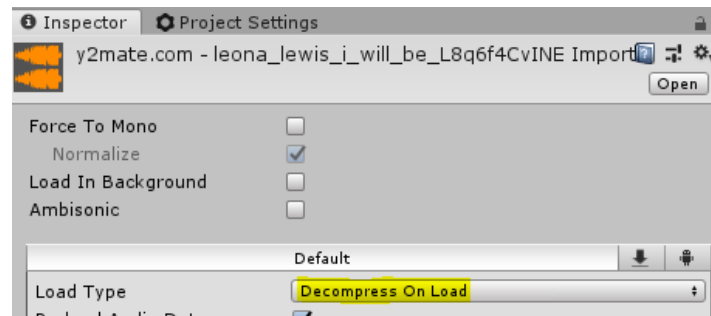


Figura 25 Descomprimir al Cargar

En el primer parámetro, float[] data, tendrá que ser del tamaño correcto, ya que guardará el total del número de muestras presentes en toda la canción. Esto quiere decir que si se tienen canciones en estéreo (como es nuestro caso), se debería añadir espacio suficiente para almacenar la información de los dos canales presentes.

Ahora queda conocer cuantas muestras hay en cada uno de los canales. La propiedad Samples, de los AudioClip indica justamente eso. Según la propia documentación de Unity, es definida como “El largo del clip de audio en samples (Read Only)”, por lo que solo quedaría multiplicar el AudioClip.samples por AudioClip.channels, el cual indica el número de canales que hay. Sabiendo el tamaño del array que se necesita para guardar el total de las samples que GetData() va a devolver, solo faltaría llamar a la función pasándole un Array[] del tamaño calculado. GetData() está devolviendo la información de los dos canales de audio en un único Array[], por lo que tiene que tener alguna forma de ordenar esta información. Lo que hace concretamente es intercalar entre la información de un canal y otro. Por lo que el Array[] devuelto quedaría de la siguiente forma:

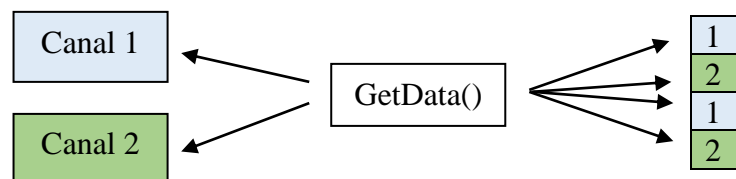


Figura 26 Canales Intercalados

Como el resultado es devuelto de forma intercalada, y el objetivo planteado es su uso en la misma forma que en el algoritmo de ritmo (donde se tenía la opción de juntar ambos canales en un solo canal mono), no se tiene más remedio que hacer esta transformación. Para conseguir este canal mono, se tendrá que realizar la media entre cada par de muestras presentes en la salida de GetData(), ya que cada par de muestras devueltas contiene la información sobre el mismo instante, pero de cada uno de los dos canales. Por esa razón, al transformar los dos canales en uno solo, se tendría en la primera posición de nuestro array lo siguiente:

$$Nuevo.Array[0] = \frac{GetDataArray[0] + GetDataArray[1]}{2}$$

Al realizar esta operación entre cada par de las samples presentes, se habría generado el nuevo Array[] mono. Nombraremos a este nuevo array como muestrasNoProcesadas, y contendrá la información de la **amplitud** en el tiempo, del audio actual. Se recalca la palabra “amplitud” para que el lector no caiga en confusión. En este array no se tendrá aun la información de la frecuencia, como se tenía antiguamente tras aplicar la FFT. Por esta razón, a este Array[] se le ha nombrado “muestrasNoProcesadas”.

Al igual que Unity facilita enormemente algunas tareas, entre ellas la facilidad con la que se puede aplicar la función FFT a un archivo de audio que se está reproduciendo en un determinado momento, no es igual de sencillo el aplicar la FFT directamente sobre la información extraída de un clip de audio. Por esta razón, en este proyecto nos hemos visto obligados a hacer uso de ciertas librerías externas, en este caso de DSPLib.

DSPLib es una librería que permite aplicar la FFT y un buen número de ventanas a una serie de datos, y lo devuelve de tal forma, que representa la misma amplitud relativa que se usaba al usar la función GetSpectrumData(). Gracias a la documentación aportada para DSPLib y el análisis de varios ejemplos que ellos mismos muestran para aprender el proceso de su uso, se pueden exponer los siguientes pasos a la hora de utilizarla:

1. Al igual que en GetSpectrumData(), ya que el proceso es el mismo que en la aplicación anterior de la FFT, es necesario dividir las muestras en una potencia de 2.
2. Iterar sobre estas divisiones para analizarlas, aplicando una ventana. En este caso se utilizará su implementación de la ventana Hanning.
3. Ejecución de la FFT sobre cada una de estas divisiones tras aplicar la ventana. Es importante tener en cuenta que la FFT devuelve números complejos y que para generar X resultados, hay que usar 2*X muestras (como se mencionó en el algoritmo de ritmo), por lo que si se quiere 1024, se tendrían que realizar divisiones de 2048. De esta forma se seguiría teniendo una precisión de 23,43kHz.
4. Transformar los números complejos en un Array[] de doubles y multiplicar por el factor de escala de Hanning para tener la amplitud correcta.

Tras haber realizado la FFT con la ayuda de la librería DSPLib, surge el problema de saber a qué tiempo corresponde cada una de las muestras que ahora se tienen almacenadas, ya que en este punto, se trata simplemente de un array[] con los datos. Si se pretende que el análisis, y los datos guardados, vayan en sincronía con la pista de audio reproduciéndose en un momento dado, hay que identificar la cantidad de tiempo que guarda cada una de las divisiones del Array[] actual.

Para calcular este tiempo hay que darse cuenta que el Array[] llamado muestrasNoProcesadas, es simplemente la información de las amplitudes de la pista, respecto al tiempo. De esta forma, cada índice de ese array, correspondía a un lapso de tiempo diferente dentro de la canción. En este caso, interesa poder transmitir a cualquiera de los algoritmos de análisis el lapso de tiempo que queremos que nos analice, y a que corresponde ese tiempo dentro del array de datos. Para ello se seguirá el siguiente proceso:

- Primeramente se calcula el número de segundos a lo que corresponde cada una de las muestras de las que se dispone. Mediante AudioClip.lenght, se obtiene la duración total de nuestro archivo de audio, y mediante AudioClip.Samples, se

obtienen el número de muestras. Por lo que solo habría que dividir la longitud en segundos, por el número de muestras.

- Mediante `audioSource.time` se puede saber el tiempo exacto que está siendo reproducido actualmente, por lo que se podrá dividir ese tiempo, por el número de segundos que corresponden a cada una de las muestras, calculado en el punto anterior.
- Una vez que se ha obtenido a que muestras corresponde el tiempo actual, se debe que recordar que estas están agrupadas en paquetes de 1024, por lo que habría que dividir por este número para obtener el índice en el `Array[]`.
- Una vez calculado este índice solo haría falta pasárselo a cualquier algoritmo de análisis para analizar la información del segundo que se está reproduciendo actualmente.

Aun habiendo conseguido el resultado deseado, pudiendo preprocesar el audio y pudiendo analizar en tiempo real, cualquier fragmento de este, uno se puede dar cuenta que el número de muestras sobre las que iterar es demasiado grande, y repercutiría demasiado en el rendimiento del juego.

La solución planteada es ejecutar estas operaciones en un hilo aparte. Para ello se hará uso de la librería de `System.Threading`, que permite exactamente esto. Unity, recomienda encarecidamente no acceder a ningún contenido de la funcionalidad de Unity API, por lo que será necesario recuperar toda la información necesaria para los futuros cálculos anteriormente, para poder pasársela al hilo. Este hilo se encargará de realizar todas las iteraciones y posteriormente devolver el resultado, para que pueda ser utilizado por el resto de los algoritmos de análisis.

Para poder comprobar que los cálculos son correctos, y el algoritmo está funcionando correctamente, se realizará una pequeña prueba. Se seleccionará una canción, por ejemplo, la canción “Hey ya” del grupo Outkast. Escuchando la canción se puede escuchar que existe un golpe de ritmo aproximadamente en el segundo 9.6-9.7. Para que el lector lo pueda apreciar correctamente, se añade el oscilograma de ese punto. Se puede ver el intervalo que se comenta por que viene “subrayado” en un gris claro.



Figura 27 Comprobación Ritmo

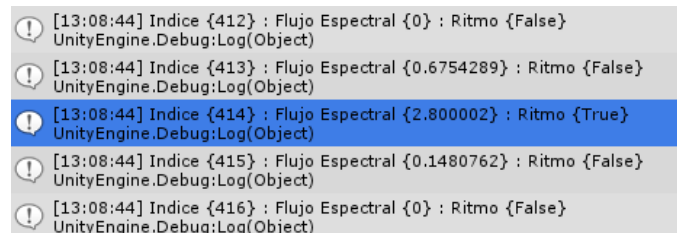
Teniendo en cuenta que esta canción tiene una frecuencia de muestro de 44100 aseguramos que:

$$\frac{1}{44100} = 2.2675 * 10^{-5}$$

Si se multiplica por el número de muestras en cada uno de los intervalos (1024) es aproximadamente 0.02321 segundos en cada intervalo.

Al querer buscar el ritmo en el segundo 9.6, se deberá que dividir 9.6s por el tiempo que hay en cada intervalo. Lo que proporciona un resultado de 413.4375. Esto quiere decir que hay que que se debería encontrar un pico alrededor de ese índice. 413-414.

Con un simple Debug.Log() en Unity de esos índices se obtiene lo siguiente:



Índice	Flujo Espectral	Ritmo
412	{0}	{False}
413	{0.6754289}	{False}
414	{2.800002}	{True}
415	{0.1480762}	{False}
416	{0}	{False}

Figura 28 Comprobación Ritmo 2

Se puede observar que, efectivamente, en el índice 414 aparece un golpe de ritmo, por lo que el preprocesamiento de señales funciona correctamente.

4.4 Analizador: Tendencia de Frecuencia

Se recuerda que una de las diferencias entre las canciones alegres y las canciones tristes es que las primeras se caracterizan, entre otras cosas, por tener una intensidad sonora más elevada. Esto quiere decir que las frecuencias presentes en el intervalo a analizar deberían contener una potencia o nivel de intensidad del sonido que indicará a que grupo pertenece. Para la realización correcta de este analizador se han utilizado los datos recogidos en nuestro Anexo 8.3 “Estudio sobre la Música”, para de este modo saber los datos esperados al analizar una u otra canción.

Gracias al preprocesamiento de audio se dispone ya de un array poblado con la información de las diferentes frecuencias desde 0Hz hasta la frecuencia de muestreo dividido entre dos. Así pues, como se tiene aplicada una longitud de 1024 a nuestro array, y la frecuencia de muestreo suele ser de 48000Hz, la frecuencia máxima devuelta tendrá un valor de 24000Hz y cada índice del array representará una cantidad aproximada de ~23Hz. Una vez dividida toda la información del espectro de la que se dispone en los 1024 índices, se procede a sacar la media de esa información. Considerando que la frecuencia que interesa esta entre 27Hz-4186Hz, no se tendrán en cuenta ciertos datos del array, ya que se trataría de ruido. Por esta razón se ignorará el primer índice del array, que contendría el intervalo de los primeros 23,43Hz. También se ignorará parte de la información guardada en los últimos índices del array, ya que se estará guardando información hasta ~24000Hz. Haciendo un simple cálculo matemático en el que se dividirán los 4186Hz que se necesitan por el tamaño de cada índice 23,43Hz, uno puede observar que como máximo será necesario hasta el índice 178 del array, que contendrá los Hz entre 4170.54Hz hasta 4193,97Hz. Se guardará entonces la información desde 23,43Hz hasta 4.193,97Hz.

Actualmente se dispone de toda la información necesaria para calcular la tendencia de las frecuencias, y se recuerda que el resultado del analizador debe ser un número entre 0 y 1. Para conseguir este resultado, se expone el siguiente enfoque. El objetivo sería hacer la media de los resultados guardados en el array, pero el array[i] actual contiene las

frecuencias exactas presentes en las muestras. Como no son relevantes los intervalos exactos si no la tendencia general de la frecuencia, podremos ignorar los Hz.

Array[i] = Intervalo Frecuencias	Intensidad Frecuencias		Índice i del Array[i]	Intensidad Frecuencias
0Hz-23,43Hz	x		0	x
23,43Hz-46,86Hz	y		1	y
46,86Hz-70,29Hz	z		2	z
--	--		--	--
i elemento	--		i	--

Tabla 2 Tendencia de Frecuencia

El siguiente paso que se realizará es obtener la media aritmética del índice del array respecto a la intensidad de frecuencias presente en cada índice. Para ello se aplicará la siguiente formula:

$$Media = \frac{\sum_{i=0}^n (i + 1) * Array[i]}{\sum_{i=0}^n Array[i]}$$

* n = índice del último elemento del array.

* y = suma de las intensidades de las frecuencias.

*(i+1) = teniendo en cuenta que el primer índice del array seria i=0, que anularía el Array[i] por el que se multiplicase.

Para poder asegurar que este cambio en las frecuencias presentes no se trata de algo puntual (como por ejemplo, el final de un estribillo o de un segmento más alegre), aprovecharemos la disponibilidad del audio completo preprocesado, para así analizar también un espacio de las muestras futuras. Mediante estas muestras futuras, y la muestra actual, se calculará la media de todas las frecuencias disponibles.

Para obtener como resultado un número del 0 al 1 que indique el nivel de tristeza-alegría que transmite la muestra analizada, se tendrá que dividir la media de cada uno de los intervalos conseguida por el número de índices en el array, teniendo en cuenta que el primer elemento no es el 0, si no el 1, por lo que:

$$\frac{Media - 1}{n}$$

Posteriormente se deberá que sumar los resultados de cada media y volverlo a dividir por el número de medias sumadas.

4.5 Analizador: Silencios y Tendencia de intensidad

En este apartado se analizará si una canción tiene o no tiene grandes diferencias en la intensidad sonora así como la media hacia donde esta tiende. Como se ha visto en el Anexo C, las canciones tristes se caracterizan por tener amplitudes bajas, con continuos intervalos de silencio en ellas.



Figura 29 Sonograma Canción Triste

Para poder localizar esta característica se calcula el coeficiente de variación de las amplitudes en las muestras de la canción. Cuya fórmula es la Desviación Típica, dividida por la media del conjunto de muestras. De esta forma es posible analizar el grado de dispersión que contiene la onda sonora.

$$C.V = \frac{Desv.Tip}{Media}$$

La desviación típica se representa por σ , y expresada por la siguiente fórmula $\rightarrow \sigma = \sqrt{\frac{\sum_i^N (X_i - \bar{X})^2}{N}}$

En cada uno de los intervalos analizados, se realizará la media de las amplitudes presentes, con las cuales, tras analizar completamente el conjunto de datos necesarios para esta característica y teniendo las medias en cada uno de los intervalos, se procederá a llevar a cabo una media de medias.

A la vez que se calculan las medias en cada uno de los intervalos en los que se itera, también se guardará el valor de sus amplitudes máximas, así como el número de intervalos analizados. De esta manera, al terminar el computo completo de todas las iteraciones, se tendrán todos los valores necesarios para poder aplicar la fórmula de la Desviación Típica. Una desviación típica menor, significará que la canción analizada transmite una emoción más triste, ya que la mayoría de los datos estarán agrupados cerca de las frecuencias bajas. Por otro lado, una mayor desviación típica, indicará que hay un mayor contraste entre amplitudes altas y bajas, como sucede en las canciones más alegres.

4.6 Contador

Como se ha mencionado en el apartado de diseño, este script se encargará de controlar el tiempo entre cada toma de muestras de la canción. De esta forma cada analizador no tendrá que llevar un continuo cálculo de su cuenta atrás (tiempo disponible), o realizar el análisis constantemente, sino que, simplemente, se usará un Evento alarm que se encargará de informarles de que el tiempo ha finalizado.

Event

Se trata de un tipo especial de dato. Con él, es posible declarar eventos. Los eventos son un suceso que se produce en un determinado momento. La ventaja que proporcionan los eventos en el presente proyecto, es que permiten que otros objetos “escuchen” determinados sucesos, y reaccionen de acuerdo a ello. Por ejemplo, en la situación actual, es posible hacer que Script TimeOut tenga un evento “alarm” que se encargue de comunicar a todos los elementos que están “escuchando”, que el tiempo ha finalizado. De esta forma, nuestros scripts de Análisis, no tienen que estar comprobando continuamente si el tiempo ha terminado, si no que al recibir el evento “alarm”, hará que se dispare determinada reacción (en este caso, comunicar el resultado).

Si se actualizase la información en los scripts de Análisis por cada frame, y teniendo en cuenta que, en una situación aceptable, un juego va a ser ejecutado a una media de 60Fps, significa que se realizaría una actualización de la información cada 0.016 segundos. Se va a cambiar esta cifra para no generar tanta sobrecarga, y en vez de ejecutar los scripts cada 0.016 veces/s, los ejecutaremos una media de 0.10. Cualquier cambio a una velocidad de 0.10 segundos resultará igualmente instantáneo, y se estará ejecutándolo unas 6 veces menos por segundo. A continuación se puede apreciar como quedaría en el Inspector de Unity.

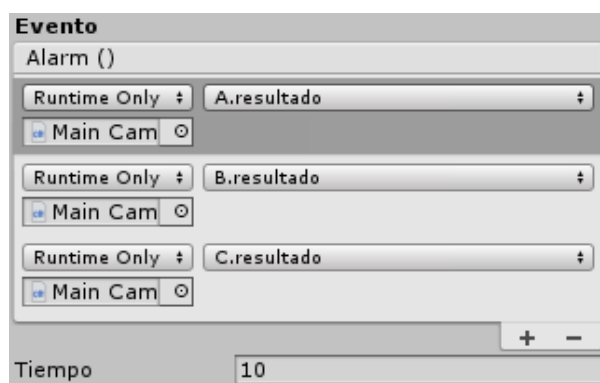


Figura 30 Contador Unity

Como se puede observar, aparecen 3 scripts (a, b, c), que equivaldrían a los diferentes scripts de análisis que se añadan (ritmo, silencios, tendencia de frecuencia...). Estos scripts se encuentran añadidos a la cámara principal, la cual, tiene el componente AudioSource. Para la adición de cualquier nuevo script, solo haría falta asociarlo a este evento, y que contenga una función (en el caso mostrado “resultado”), que transmita los cálculos efectuados al Script “Controlador_resultados”. Por último, podemos observar que existe una variable llamada tiempo, donde se indicará cada cuanto se ejecutará el Evento alarm. El resultado sería entonces un Script “Contador”, que ejecutaría el Event “alarm” cada cierto intervalo de tiempo. Para realizar esto, se usará la función:

```
InvokeRepeating(nombreFunción , segundosInicio, intervaloTiempo)
```

Esta función permitirá ejecutar el Evento alarma en “segundosInicio” segundos, y luego ejecutarla cada “intervaloTiempo” segundos.

4.7 Controlador_resultados

Para almacenar todos los datos proporcionados por los diferentes scripts de Análisis en el script Controlador_resultados se ha optado por declarar un List<Vector2> listaValores. De esta forma, se podrá ir añadiendo resultados sin saber el número de scripts de análisis que se tiene en su totalidad en ese determinado momento, y por lo tanto la cantidad de información que van a llegar. Se ha escogido un Vector2 como el tipo de datos que contendrá la lista, teniendo en cuenta que cada resultado enviado, estará formado por dos valores:

- X: Multiplicador del resultado enviado.
- Y: Resultado enviado entre 0-1. *

Por lo tanto, desde cada script de Análisis, se dará un determinado peso a la información que estamos enviando. Por ejemplo, si el ritmo es menos determinante que la frecuencia presente en una canción, el multiplicador del ritmo, será menor que el de la frecuencia.

*Hay que tener en cuenta que como los resultados representan cosas totalmente diferentes y se necesita que todos tengan cierto equilibrio para conseguir un resultado lógico, cada script de Análisis se encargará de acotar el resultado entre 0 y 1. Siendo 0 el resultado más triste y 1 el resultado más alegre. Si no se tomase esta medida se podría llegar a tener, por ejemplo, resultados de ritmo entre 0-100 y de frecuencia media entre 27-4000.

Una vez ha sido enviada la información de cada analizador, Controlador_resultados irá recorriendo la lista y almacenando la suma de cada resultado por su multiplicador de la siguiente manera: $\text{ResultadoTotal} += \text{multiplicador} * \text{resultado}$. Al final, se tendrá un número (ResultadoTotal). El resultado definirá el nivel emocional de la canción en ese intervalo, tras ser procesada por todos los scripts de Análisis disponibles. Gracias a esta información, los diferentes efectos que se implementen en la escena se comportarán de una u otra manera, pudiendo regular su intensidad en función de este ResultadoTotal.

5 Integración, pruebas y resultados

Durante el desarrollo de este Trabajo de Fin de Grado se han ido realizando pequeñas pruebas de los algoritmos que en cada apartado diseñado. Se puede ver una de estas pruebas en la comprobación sobre el algoritmo de ritmo, en el que se busca un ritmo de una canción de forma manual para verificar que la FFT estaba siendo realizada correctamente.

Una vez comprobado el funcionamiento de la FFT, es indispensable comprobar también el correcto valor de los diferentes analizadores. Hay que recordar que el objetivo es poder puntuar las diferentes canciones según el estado emocional que generen en el oyente. Para este fin se tienen los resultados de nuestro “Estudio sobre la música”, con el que nos guiaremos a la hora de comprobar los valores devueltos por los algoritmos desarrollados, y ver si corresponden con los resultados esperados de nuestro estudio.

Los resultados esperados se tratan de aproximaciones generales que se han obtenido gracias al sondeo de un grupo de personas. En las pruebas realizadas a continuación pueden salir resultados que no sean totalmente acordes con los esperados inicialmente. Esto se debe a que, por una parte, se están realizando pruebas individuales a características específicas de cada canción, que pueden no influir en la emoción que generan tanto como se esperaría en un principio, y por otra parte puede deberse a que muchas canciones tienen características que no siguen el esquema, que innovan y que no se pueden asociar directamente con ciertas características comunes. Igualmente, como se verá en los siguientes párrafos, en la mayoría de los casos se puede ver una tendencia general correcta de las canciones respecto a los resultados esperados en cada una de las características analizadas.

Analizador de Ritmo: Podemos comprobar la precisión del algoritmo comparando los resultados obtenidos frente a los devueltos por las diferentes páginas web que se dedican a este fin, como son:

- getsongbpm
- tunebat
- songbpm

Por ejemplo, para la canción “Don’t Stop Me Now” del grupo Queen, se espera un BPM de 156 según el estudio realizado, y en el algoritmo de ritmo devuelve un BPM de:

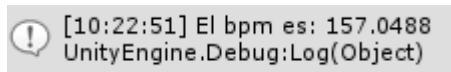


Figura 31 Resultado BPM

La comparación entre estos dos datos, nos da un Error Absoluto (Ea) de 1.0488 bpms, y un Error Relativo (Er) de un 0.67 %.

Las salidas obtenidas varían ligeramente de una canción a otra. Por ejemplo, en la canción “walking on sunshine” se tiene un Er de 13.11%. Bastante mayor al visto anteriormente. Por otra parte, si se prueba con canciones más tristes, donde la precisión suele ser menor,

se pueden obtener errores ligeramente mayores. Esto se puede ver analizando la canción “The last goodbye” de billy boyd, donde alcanza un Er de casi el 20%.

En general, aun tratándose el análisis de BPM de un proceso complicado, se obtienen errores entre un 0% en los mejores casos, y un 20% en los peores.

A continuación puede observarse el grafico de los resultados, donde estaban ordenadas las canciones de más tristes (izquierda), a más alegres (derecha).

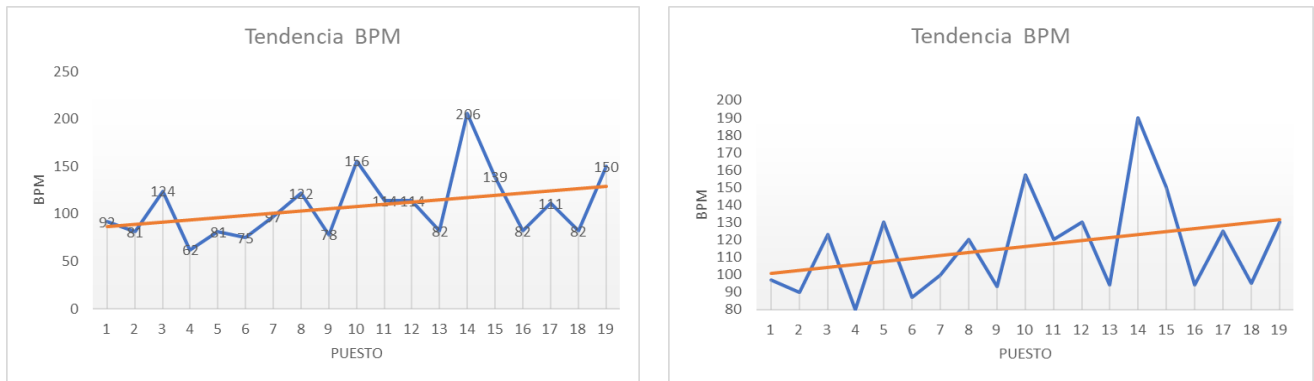


Figura 32 Resultado del ritmo

La comparación entre el resultado del estudio (imagen izquierda), con el resultado que se obtiene de Unity (derecha) nos permite observar que hay una gran similitud.

Analizador de Intensidad: Se recuerda que los grandes niveles de intensidad eran representativos de las canciones alegres, mientras que las canciones tristes solían tener una media de intensidad mucho menor.



Figura 33 Resultado Intensidad

Los datos obtenidos por nuestro analizador que encajan perfectamente con la hipótesis planteada. En este gráfico se puede observar como a medida que la canción es considerada más alegre (hacia la derecha en el eje horizontal), la intensidad aumenta de forma gradual.

Analizador de Frecuencia: La hipótesis inicial en este algoritmo era que en las canciones consideradas como alegres, existía una mayor proporción de frecuencias altas, frente a las canciones consideradas como tristes. A continuación se podrá observar la gráfica generada por nuestro algoritmo, la cual, coincide totalmente con los resultados esperados.

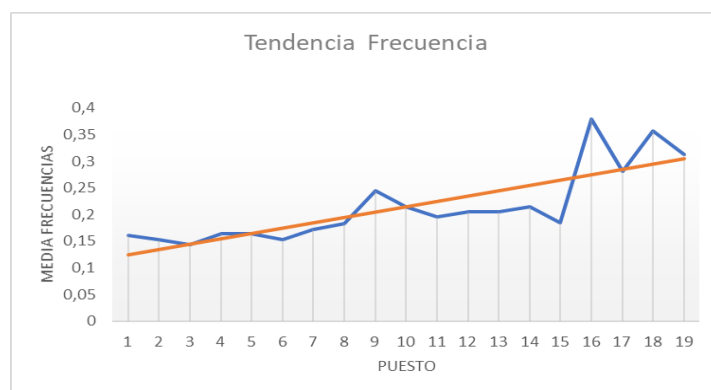


Figura 34 Resultado Frecuencia

Por último, queda comprobar los resultados globales, obtenidos de la combinación de la totalidad de resultados parciales de los algoritmos teniendo en cuenta las ponderaciones de cada uno de ellos. Gracias a esto, podremos confirmar que con los datos obtenidos en cada análisis realizado se va a poder valorar las canciones de forma correcta y asociarlas a una determinada atmosfera emocional.

El resultado final obtenido que llega a “Controlador_resultados” (recordemos que es el que se va a encargar de controlar las respuestas a los resultados obtenidos) es el siguiente:



Figura 35 Resultado Final

En este resultado se puede observar claramente la utilidad del proyecto. Se aprecia una línea ascendente que encaja a la perfección con los resultados esperados, y que servirá para poder generar los diferentes efectos (niebla, lluvia, etc.) creados en respuesta a estos datos. Existen pequeños picos en la gráfica general, como en la canción 14,9 y 8 que se salen ligeramente de la tendencia. Estos datos no suponen mayor problema, ya que lo que nos interesa a nosotros es la posibilidad de poder ordenar aproximadamente los resultados generales, y corroborar que, efectivamente, las salidas de los algoritmos concuerdan con las hipótesis planteadas en el Anexo C, “Estudio sobre la música”.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

En este apartado se expondrán las diferentes conclusiones que se han ido recopilando durante el desarrollo de este Trabajo de Fin de Grado.

Al final de este proyecto se presenta un resultado capaz de analizar las diferentes canciones que el usuario quiera utilizar en su propio juego, para poder así interpretarlas y saber qué clase de emoción van a evocar en el jugador. La solución planteada presenta un alto acierto en el análisis de ciertos géneros musicales, pero peca de falta de precisión a la hora de analizar otros. Es lógico pensar que hay canciones que distan mucho unas entre sí, y sus características también son totalmente diferentes, de ahí surge la complicación de poder analizarlas todas de forma eficiente aun teniendo en cuenta la disparidad que presentan entre ellas. Además, gracias a la incorporación del uso de variables para la personalización de la sensibilidad, es sencillo minimizar ciertos errores o dificultades encontrados a la hora de analizar una canción. Estas dificultades se han podido apreciar por ejemplo, a la hora de analizar el ritmo de una canción, donde en las canciones tristes resultaba mucho más difícil debido a su falta de énfasis en los BPM.

Respecto al rendimiento del proyecto desarrollado, es apto para poder ejecutarse en cualquier ordenador sin que requiera unas grandes prestaciones, ya que gracias a la realización de las operaciones más pesadas en un hilo a parte del proceso principal, no se producen caídas de rendimiento. Todas las pruebas han sido testeadas en un ordenador personal, un portátil de gama media-baja, sin ningún tipo de problema.

En términos generales la generación de un algoritmo capaz de analizar las emociones transmitidas por un audio, es una herramienta diferente de enorme utilidad que puede dar lugar a un gran número de posibilidades diferentes. No está limitado a la sinergia de la canción con la atmósfera o el clima del entorno que le rodea, si no que puede servir para hacer que ciertos enemigos actúen de una forma diferente según la música, o el propio escenario, o incontables mecánicas más, solo limitadas por la imaginación de cada uno.

6.2 Trabajo futuro

En la conclusión se ha puesto de manifiesto el potencial que tiene este proyecto a la hora de crear juegos dinámicos y originales, aunque bien es cierto que tiene ciertos matices que pulir para mejorar su funcionamiento y precisión general, por lo que se han propuesto algunas medidas para mejorarlo. Algunas de estas propuestas son:

- Pulir las transiciones entre unos efectos y otros. Cuando la música cambie repentinamente a otro tipo de emoción, debería existir una transición más fluida.
- Mejorar la precisión del proyecto en ciertos géneros musicales.
- Mejorar el rendimiento puliendo la selección de umbrales y frecuencias a la hora de realizar las diferentes comprobaciones, medias, desviaciones típicas, etc.
- Aumentar el número de efectos ambientales desarrollados (niebla, lluvia, etc.).

7 Referencias

- [Referencia 1]. Pablo Lezana Illesca y Jorge Muratt Rodríguez. “Transformada rápida de Fourier FFT”, Lectura 5 de la universidad técnica de Federico Santa María departamento de electrónica. 10, mayo 2005.
- [Referencia 2]. W. M. Gentleman and G. Sande. “Fast Fourier Transforms – For Fun and Profit”. Disponible en: http://cis.rit.edu/class/simg716/FFT_Fun_Profit.pdf. Princeton University. 578 págs. Noviembre 1966.
- [Referencia 3]. Ángel Franco García. “Transformada Rápida de Fourier (I)”, Disponible en: http://www.sc.ehu.es/sbweb/fisica3/datos/fourier/fourier_1.html. Universidad del País Vasco. 2016.
- [Referencia 4]. Daniela Moscoso. “La música”. Disponible en: <http://danielamoscoso1b.blogspot.com/>. Abril 2015.
- [Referencia 5]. Julián Pérez Porto y Ana Gardey. “Definición de sonido”. Disponible en: <https://definicion.de/sonido/>. publicado 2010 y actualizado 2012.
- [Referencia 6]. Pablo Fernández-Cid. “Armónicos y parciales, concepto del fundamental”. Disponible en: <https://www.hispasonic.com/tutoriales/sinusoides-sonidos-compuestos-armonicos-parciales-audio/43309#section5>. Noviembre 2017.
- [Referencia 7]. Santiago Vilanova Ángeles. “Análisis de Audio”. Disponible en: http://docplayer.es/6733555-Analisis-de-audio-santiago-vilanova-angeles-pid_00184754.html. Universitat Oberta de Catalunya. 22 págs. 2015.
- [Referencia 8]. Jimmy Alexander Cortes Osorio and Andrew M. Knott and José Andrés Chaves Osorio. “Aproximación a la síntesis de la música a través del análisis de Fourier”. “An approach to the synthesis of music through Fourier Analysis”. Departamento de Física, Universidad Tecnológica de Pereira, Pereira, Colombia. 7 págs. Diciembre 2012.
- [Referencia 9]. Gil Fernández. “Los sonidos del lenguaje: Visualización de las características acústicas del habla en los documentos del análisis acústico”. Editorial Síntesis. 44 págs. 2014.
- [Referencia 10]. Lía Pérez. “Los armónicos”. Disponible en: <https://www.monografias.com/trabajos59/los-armonicos/los-armonicos.shtml>. 2 págs. Agosto 2008.
- [Referencia 11]. Gustavo Basso. “Análisis Espectral. La transformada de Fourier en la Música”. Colección Universitaria -La Plata-. 198 págs. 2001.
- [Referencia 12]. Rémi Pujol, María Morell y Pablo Gil-Loyzaga. “Campo Auditivo Humano”. Disponible en: <http://www.cochlea.org/es/sonidos/campo-auditivo-humano>. Junio 2018.
- [Referencia 13]. Enrique Soto. “El oído y la música”. Disponible en: <https://www.monografias.com/trabajos918/oído-musica/oído-musica.shtml>. 2 págs. Marzo 2019.
- [Referencia 14]. Pedro Tovar. “Las 7 características principales de la música”. Disponible en: <https://www.lifeder.com/caracteristicas-musica/>.
- [Referencia 15]. José Luis Díaz. “Music, language and emotion: a brain approach”. 553-551. 2010.
- [Referencia 16]. María de Sancha Rojo. “Así se juega con la música para provocar alegría, calma, melancolía o tristeza”. Disponible en: https://www.huffingtonpost.es/2017/03/04/asi-se-juega-con-la-musica-para-provocar-alegria-calma-melanco_a_21872596/. Marzo 2017.

- [Referencia 17]. Juan Roederer. "The Physics and Psychophysics of Music". Springer-Verlag. 198 págs. 1995.
- [Referencia 18]. Stanley A. Gelfand. "An Introduction to Psychological and Physiological Acoustics". 322 págs. 2010.
- [Referencia 19]. Brian C. J. Moore. "An Introduction to the Psychology of Hearing ". Department of Experimental Psychology, University of Cambridge. 457 págs. 2013.
- [Referencia 20]. Sebastián Miranda Payacán. "¿Psicología y música?". Disponible en: <https://www.psyciencia.com/psicologia-y-musica/>. Septiembre 2012.
- [Referencia 21]. Eugenio Enríquez, Ingrid Gómez, Claudia Rojas y Julia Libreros. "Psicología de la Música". Disponible en: <https://es.slideshare.net/eugenioenriquez2011/psicologia-de-la-musica>. 30 págs. marzo 2012.
- [Referencia 22]. Mario Zechner. "Beat Detection – The Scientific View". Disponible en: <https://www.badlogicgames.com/wordpress/?p=99>. Febrero 2010.
- [Referencia 23]. Jonathan Foote y Shingo Uchihashi. "The beat spectrum: A new approach to rhythm analysis". Disponible en: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.8485&rep=rep1&type=pdf>. 4 págs. Agosto 2001.
- [Referencia 24]. Scheirer, E. "Tempo and Beat Analysis of Acoustic Musical Signals," in J. Acoust. 601 págs. Enero 1998.
- [Referencia 25]. Mario Zechner. Disponible en: <https://www.badlogicgames.com/wordpress/?cat=18>. Febrero 2010.
- [Referencia 26]. Joaquim Llisterri. "Métodos de análisis acústico del habla". Departament de Filologia Espanyola, Universitat Autònoma de Barcelona . http://liceu.uab.es/~joaquim/phonetics/fon_anal_acus/met_anal_acust.html. Abril 2019.
- [Referencia 27]. José María Hernández de Miguel. "Estructura y características de la señal acústica, estudio y caracterización". Disponible en: <https://jmhweb.wordpress.com/docencia/master-en-biologia-de-la-conservacion-ucm/estructura-y-caracteristicas-de-la-senal-acustica-estudio-y-caracterizacion/>. Noviembre 2012.

8 Anexos

8.1 Manual de instalación

Para su correcto uso solo es necesario cargar el script de Controlador_fourier en un GameObject que contenga un componente AudioSource junto con los diferentes scripts de análisis. Por otra parte, se añadirá otro GameObject llamado MusicController donde irán los scripts de Contador y Controlador_resultados. Finalmente solo hará falta añadir una canción al AudioSource y ejecutar la escena.

8.2 Manual del programador

Para la adición de analizadores bastará con hacer uso del Evento definido en la clase contador, donde se podrán añadir otros analizadores encargados de estudiar características diferentes de la canción. Únicamente será necesario que el script que se añada tenga un método llamado resultado() que se encargue de devolver su resultado y ponderación, y de esta forma comunicárselo a Controlador_resultados.

8.3 Estudio sobre la música

Tratándose la música, como hemos mencionado anteriormente, de un tema principalmente cultural, se ha decidido realizar un pequeño estudio con 20 participantes, que nos ayudará a esclarecer la opinión general sobre los sentimientos que genera una u otra canción.

Con los datos que obtengamos de realizar el estudio, podremos realizar una comprensión más correcta de las características que poseen las canciones consideradas como tristes, y las canciones consideradas como alegres, así como los límites que separan unas de otras.

Para la realización de este estudio se usarán 15 canciones variadas que se mostrarán más adelante. Se han separado las canciones en 3 grupos diferentes. Las canciones de cada grupo tienen ciertas características similares entre ellas, como puede ser el bpm, la frecuencia media, la máxima, etc. Por un lado se tiene el grupo de las canciones consideradas “tristes”, por otro lado las canciones “alegres”, y luego un tercer grupo que tiene canciones que no son canciones ni alegres ni tristes, si no un punto intermedio. Con este grupo intermedio podremos ver el punto exacto en el que los participantes empiezan a considerar si una canción tiende más a la alegría o a la tristeza, y de esta forma poder observar la tendencia de las características que tiene la canción en cada caso.

Gracias a los datos que recabados en este estudio se podrán implementar de forma más correcta los algoritmos necesarios para poder diferenciar los sentimientos que provocan unas canciones respecto a otras, ya que, clasificando las canciones correctamente, podremos analizar con más profundidad las diferencias que tienen unas respecto a las otras (intensidad, frecuencias, ritmo...), incluso descubrir características nuevas que pueden diferenciarlas y que hasta ahora no habíamos tenido en cuenta. El objetivo final no es solo encontrar esas características que más influyen a un oyente, así como distinguir las

diferentes características de las canciones mencionadas anteriormente, si no también, encontrar los valores aproximados en los que una canción provoca un determinado sentimiento, u otro muy diferente. Esto nos será muy útil a la hora de manejar los valores de las intensidades y de las frecuentes presentes en cada sonido.

Lo que se les ha pedido a los participantes es que ordenen las 15 obras musicales diferentes según lo alegre que las consideren, recalcando el punto intermedio en el que las canciones no les provocan una sensación alegre, ni triste. De esta manera no se tendrá que basar el desarrollo de nuestro algoritmo en únicamente nuestra propia opinión, si no que tendremos una idea más general y más correcta.

Las 15 canciones seleccionadas son las siguientes:

Título (Autor)	Núm.
Let her go (Passenger)	1
Hey ya! (Outkast)	2
You get what you give (New Radicals)	3
I can feel a hot one (Manchester Orchestra)	4
The last goodbye (Billy Boyd)	5
Feel good inc (Gorillaz)	6
High Hopes (Panic! At The Disco)	7
The end of Love (Florence And The Machine)	8
Palace (Sam Smith)	9
Eat Sleep Love you repeat (Rodney Atkins)	10
Dont stop me now (Queen)	11
Cats in the Cradle (Harry Chapin)	12
Walking on Sunshine (Katrina And The Waves)	13
Tired of waiting for you (The Kinks)	14
Roundabouts (Michael Patrick Kelly)	15

Tabla 3 Canciones Estudio

A cada canción se le ha dotado de un número para simplificar su escritura en esta memoria. De esta forma, de ahora en adelante, se nombrará cada a cada una por su número asociado, y no por el nombre de la pieza musical. Por otra parte, a los participantes se les ha entregado un archivo que contenía las 15 canciones con únicamente sus números asociados, de esta forma se evita que dispongan de cualquier información adicional que pueda influir en su decisión (como los autores de las mismas).

Los resultados del sondeo son los siguientes:

	Número de la canción - —————> +														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
↓ Nivel de alegría	1	0	0	0	6	10	0	0	2	2	0	0	0	0	0
	2	2	0	0	2	0	0	0	6	10	0	0	0	0	0
	3	3	0	0	6	3	0	0	0	3	0	0	0	0	3
	4	3	0	0	3	0	0	0	6	6	0	0	0	0	2
	5	7	0	0	2	2	0	0	3	0	0	0	3	0	0
	6	0	0	0	0	3	3	0	3	0	0	0	3	0	0
	7	3	0	3	0	0	3	0	0	0	0	0	3	0	8
	8	0	0	0	0	0	3	0	0	0	0	0	11	0	3
	9	2	0	2	0	0	0	0	0	0	3	7	0	0	3
	10	0	0	9	0	2	0	0	0	0	6	3	0	0	0
	11	0	0	6	0	0	0	6	0	0	6	0	0	2	0
	12	0	0	0	0	0	10	0	0	0	0	5	0	5	0
	13	0	2	0	0	0	0	6	0	0	3	3	0	6	0
	14	0	3	0	0	0	3	6	0	0	3	3	0	2	0
	15	0	14	0	0	0	0	3	0	0	0	0	0	3	0

Tabla 4 Tabla Resultados

En esta tabla se pueden ver resumidas las opiniones de las 20 personas que han participado en el sondeo. En el eje horizontal están las canciones, representadas por su número (15 canciones disponibles), mientras que en el eje vertical encontramos ordenados del 1 al 15 el puesto que ocupan en la lista general (siendo el puesto 1 el correspondiente a la canción más triste, y el número 15 a la canción más alegre).

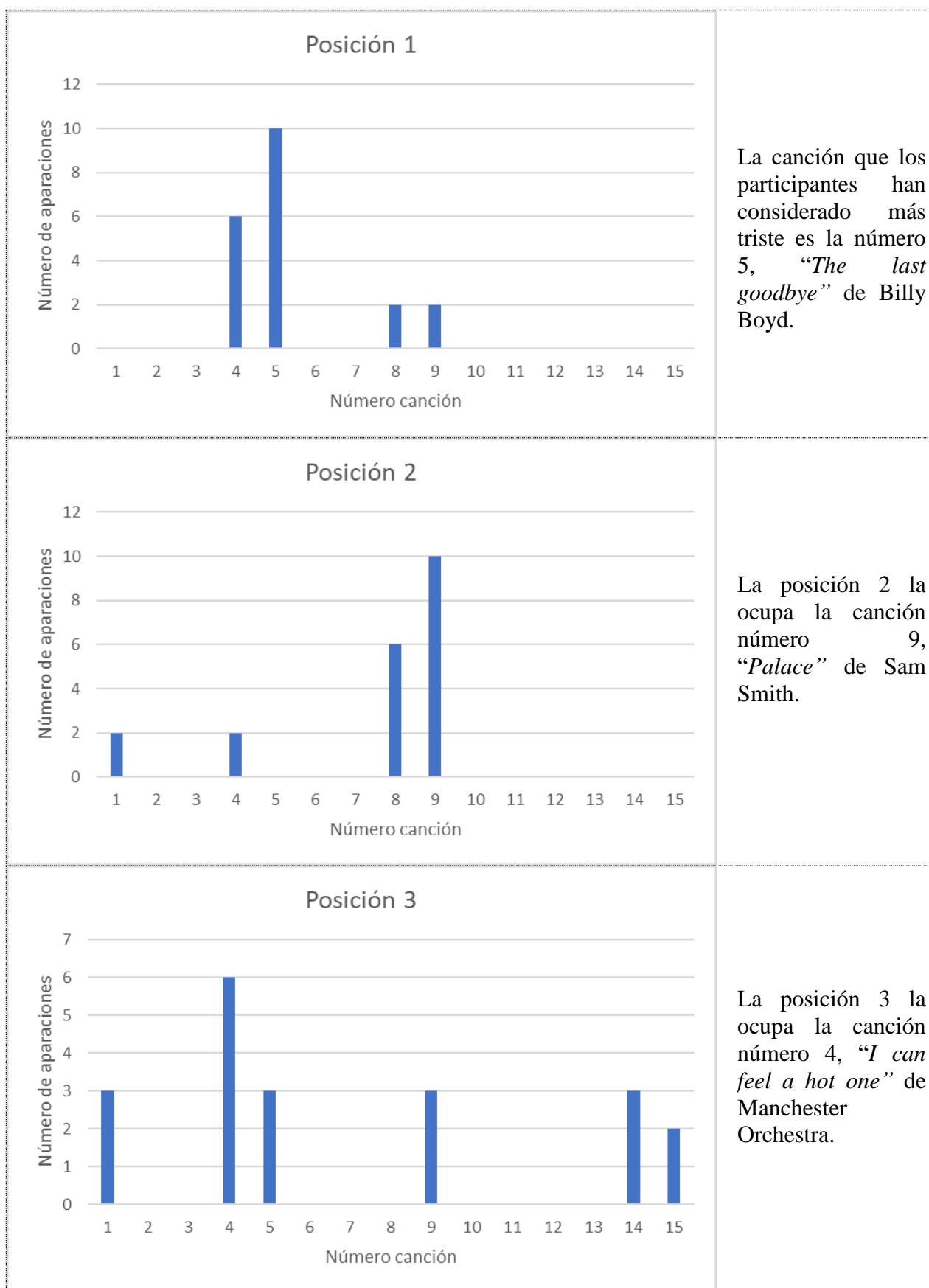
Por ejemplo, en la posición número 1, podemos ver los resultados:

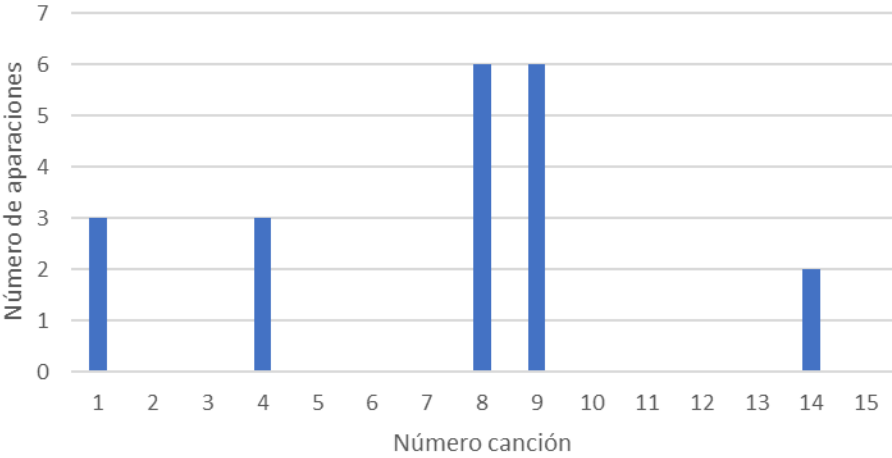
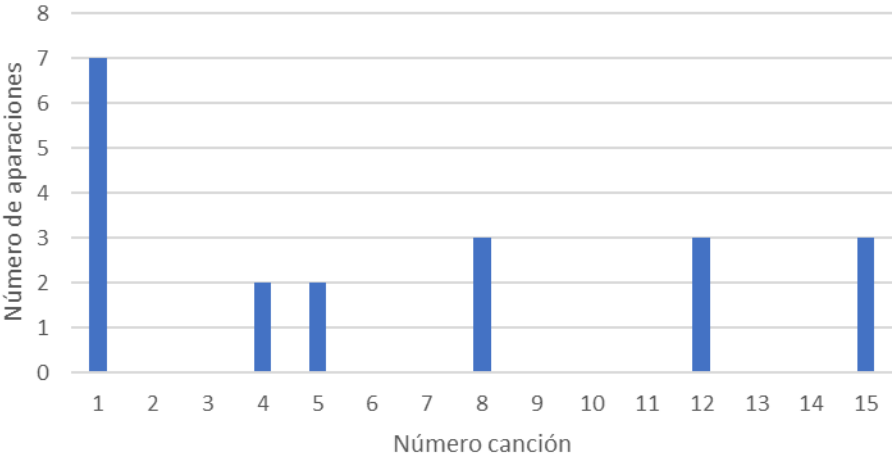
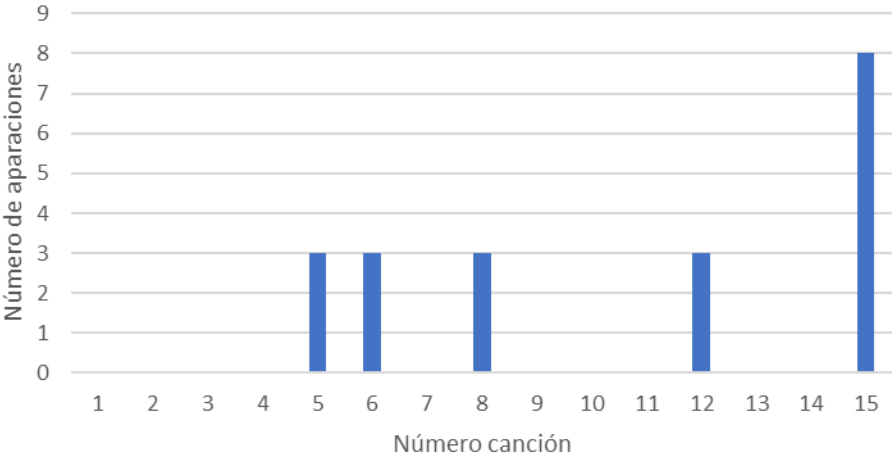
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0 0 0 6 10 0 0 2 2 0 0 0 0 0 0

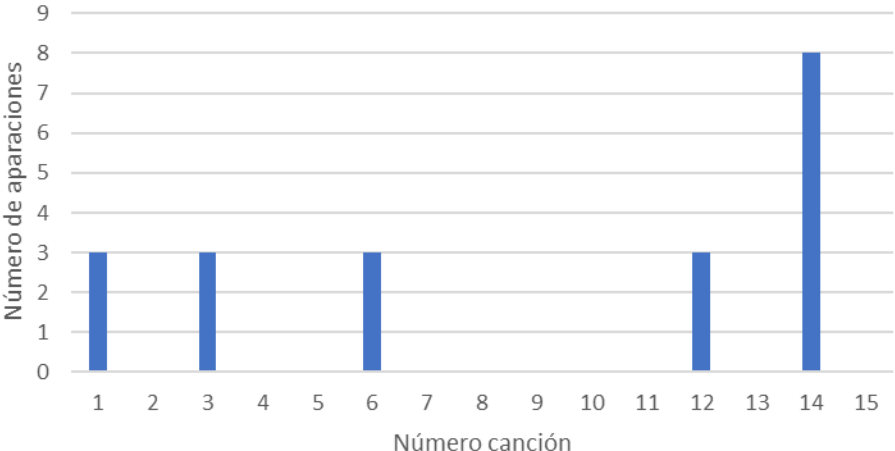
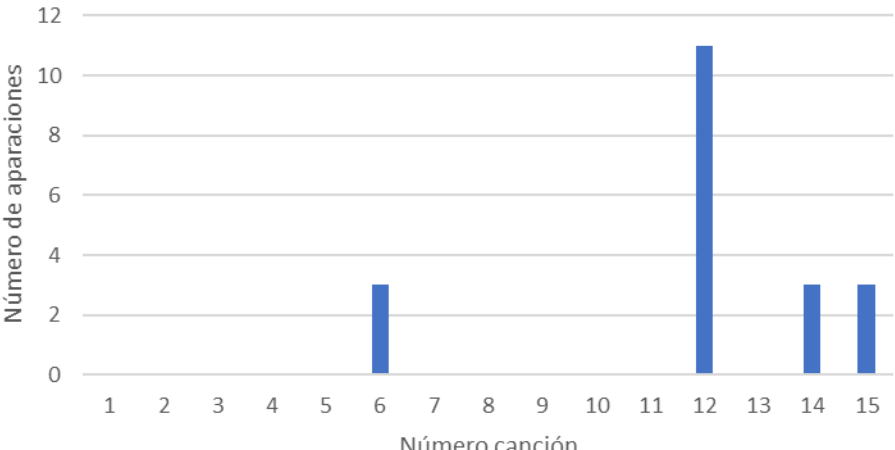
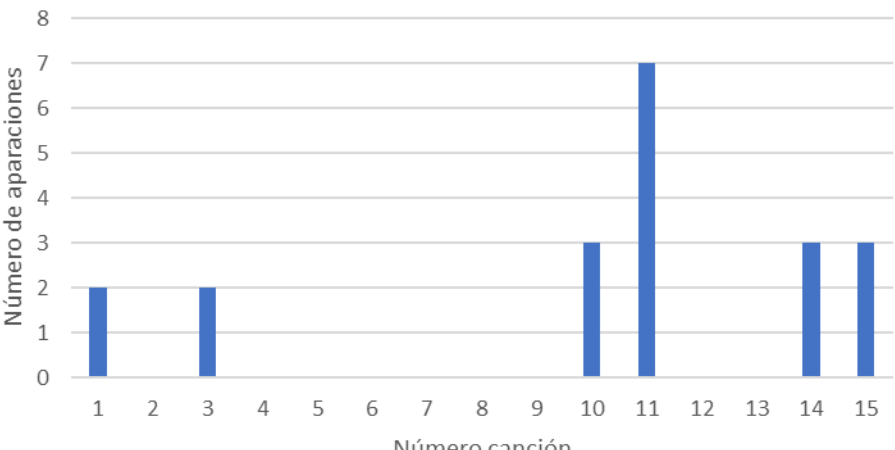
Eso significa que 6 de los participantes han considerado que la canción número 4 es la canción más triste de todas, 10 participantes creen que la canción más triste es la número 5, 2 participantes han escogido la número 8 y otros 2 creen que es la número 9. En total, en cada posición hay 20 votos, correspondiente a los 20 participantes.

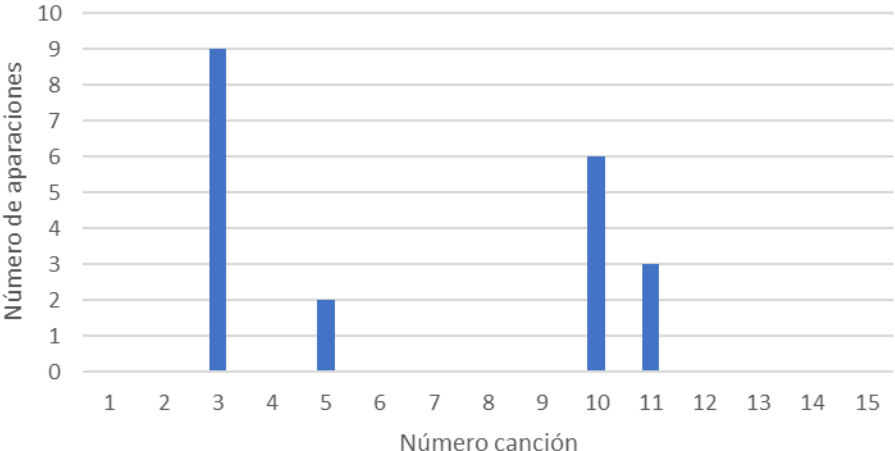
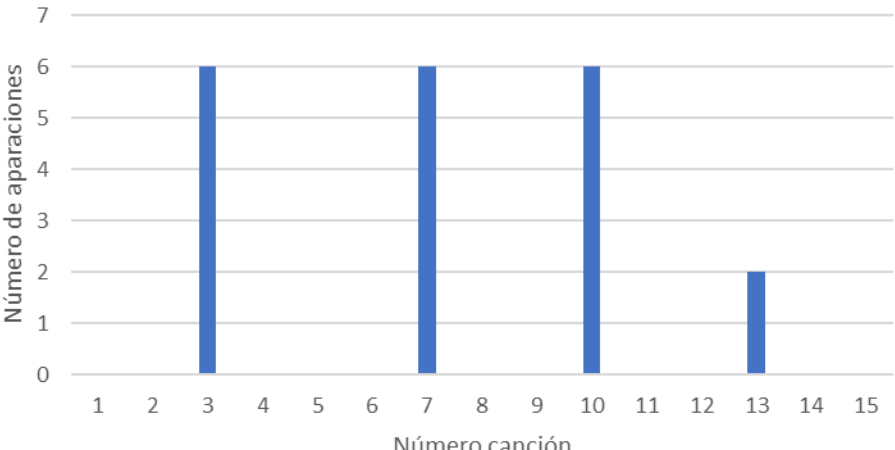
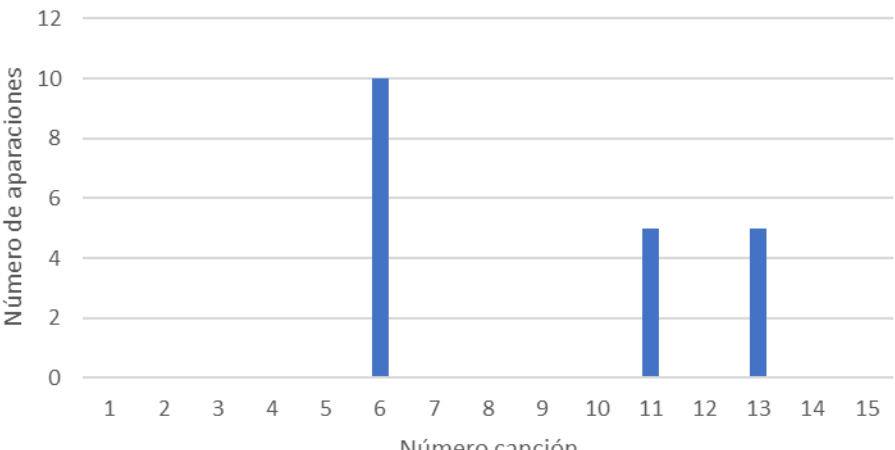
A continuación, podremos ver unas gráficas correspondientes a cada puesto del ranking (1 al 15) y las canciones que los participantes han colocado más en cada uno.

Ordenados según la media de la posición en la que los 20 participantes han ordenado las canciones, tratándose la posición 1 de la canción más triste, y la posición 15 de la más alegre



<p style="text-align: center;">Posición 4</p>  <table border="1"> <thead> <tr> <th>Número canción</th> <th>Número de apariciones</th> </tr> </thead> <tbody> <tr><td>1</td><td>3</td></tr> <tr><td>4</td><td>3</td></tr> <tr><td>8</td><td>6</td></tr> <tr><td>9</td><td>6</td></tr> <tr><td>14</td><td>2</td></tr> </tbody> </table>	Número canción	Número de apariciones	1	3	4	3	8	6	9	6	14	2	<p>La posición 4 la ocupa la canción número 8 y la número 9, “<i>The end of Love</i>” de Florence and the Machine y “<i>Palace</i>” de Sam Smith.</p>		
Número canción	Número de apariciones														
1	3														
4	3														
8	6														
9	6														
14	2														
<p style="text-align: center;">Posición 5</p>  <table border="1"> <thead> <tr> <th>Número canción</th> <th>Número de apariciones</th> </tr> </thead> <tbody> <tr><td>1</td><td>7</td></tr> <tr><td>4</td><td>2</td></tr> <tr><td>5</td><td>2</td></tr> <tr><td>8</td><td>3</td></tr> <tr><td>12</td><td>3</td></tr> <tr><td>15</td><td>3</td></tr> </tbody> </table>	Número canción	Número de apariciones	1	7	4	2	5	2	8	3	12	3	15	3	<p>La posición 5 la ocupa la canción número 1, “<i>Let her go</i>” de Passenger.</p>
Número canción	Número de apariciones														
1	7														
4	2														
5	2														
8	3														
12	3														
15	3														
<p style="text-align: center;">Posición 6</p>  <table border="1"> <thead> <tr> <th>Número canción</th> <th>Número de apariciones</th> </tr> </thead> <tbody> <tr><td>5</td><td>3</td></tr> <tr><td>6</td><td>3</td></tr> <tr><td>8</td><td>3</td></tr> <tr><td>12</td><td>3</td></tr> <tr><td>15</td><td>8</td></tr> </tbody> </table>	Número canción	Número de apariciones	5	3	6	3	8	3	12	3	15	8	<p>La posición 6 la ocupa la canción número 15, “<i>Roundabouts</i>” de Michael Patrick Kelly.</p>		
Número canción	Número de apariciones														
5	3														
6	3														
8	3														
12	3														
15	8														

<p style="text-align: center;">Posición 7</p>  <table border="1"> <thead> <tr> <th>Número canción</th> <th>Número de apariciones</th> </tr> </thead> <tbody> <tr><td>1</td><td>3</td></tr> <tr><td>3</td><td>3</td></tr> <tr><td>6</td><td>3</td></tr> <tr><td>12</td><td>3</td></tr> <tr><td>14</td><td>8</td></tr> </tbody> </table>	Número canción	Número de apariciones	1	3	3	3	6	3	12	3	14	8	<p>La posición 7 la ocupa la canción número 14, “<i>Tired of waiting for you</i>” de The Kinks.</p>		
Número canción	Número de apariciones														
1	3														
3	3														
6	3														
12	3														
14	8														
<p style="text-align: center;">Posición 8</p>  <table border="1"> <thead> <tr> <th>Número canción</th> <th>Número de apariciones</th> </tr> </thead> <tbody> <tr><td>6</td><td>3</td></tr> <tr><td>12</td><td>11</td></tr> <tr><td>14</td><td>3</td></tr> <tr><td>15</td><td>3</td></tr> </tbody> </table>	Número canción	Número de apariciones	6	3	12	11	14	3	15	3	<p>La posición 8 la ocupa la canción número 12, “<i>Cats in the Cradle</i>” de Harry Chapin.</p>				
Número canción	Número de apariciones														
6	3														
12	11														
14	3														
15	3														
<p style="text-align: center;">Posición 9</p>  <table border="1"> <thead> <tr> <th>Número canción</th> <th>Número de apariciones</th> </tr> </thead> <tbody> <tr><td>1</td><td>2</td></tr> <tr><td>3</td><td>2</td></tr> <tr><td>10</td><td>3</td></tr> <tr><td>11</td><td>7</td></tr> <tr><td>14</td><td>3</td></tr> <tr><td>15</td><td>3</td></tr> </tbody> </table>	Número canción	Número de apariciones	1	2	3	2	10	3	11	7	14	3	15	3	<p>La posición 9 la ocupa la canción número 11, “<i>Dont stop me now</i>” de Queen.</p>
Número canción	Número de apariciones														
1	2														
3	2														
10	3														
11	7														
14	3														
15	3														

<p>Posición 10</p>  <table border="1"><thead><tr><th>Número canción</th><th>Número de apariciones</th></tr></thead><tbody><tr><td>3</td><td>9</td></tr><tr><td>5</td><td>2</td></tr><tr><td>10</td><td>6</td></tr><tr><td>11</td><td>3</td></tr></tbody></table>	Número canción	Número de apariciones	3	9	5	2	10	6	11	3	<p>La posición 10 la ocupa la canción número 3, <i>“You get what you give”</i> de New Radicals.</p>
Número canción	Número de apariciones										
3	9										
5	2										
10	6										
11	3										
<p>Posición 11</p>  <table border="1"><thead><tr><th>Número canción</th><th>Número de apariciones</th></tr></thead><tbody><tr><td>3</td><td>6</td></tr><tr><td>7</td><td>6</td></tr><tr><td>10</td><td>6</td></tr><tr><td>13</td><td>2</td></tr></tbody></table>	Número canción	Número de apariciones	3	6	7	6	10	6	13	2	<p>La posición 11 la ocupa la canción número 3, la número 7 y la número 10, <i>“You get what you give”</i> de New Radicals, <i>“High Hopes”</i> de Panic! At The Disco, y <i>“Eat Sleep Love you repeat”</i> de Rodney Atkins.</p>
Número canción	Número de apariciones										
3	6										
7	6										
10	6										
13	2										
<p>Posición 12</p>  <table border="1"><thead><tr><th>Número canción</th><th>Número de apariciones</th></tr></thead><tbody><tr><td>6</td><td>10</td></tr><tr><td>11</td><td>5</td></tr><tr><td>13</td><td>5</td></tr></tbody></table>	Número canción	Número de apariciones	6	10	11	5	13	5	<p>La posición 12 la ocupa la canción número 6, <i>“Feel good inc”</i> de Gorillaz.</p>		
Número canción	Número de apariciones										
6	10										
11	5										
13	5										

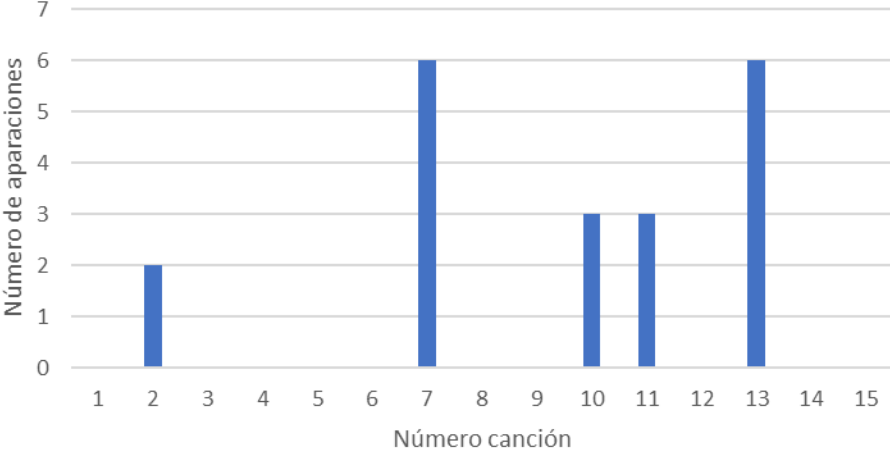
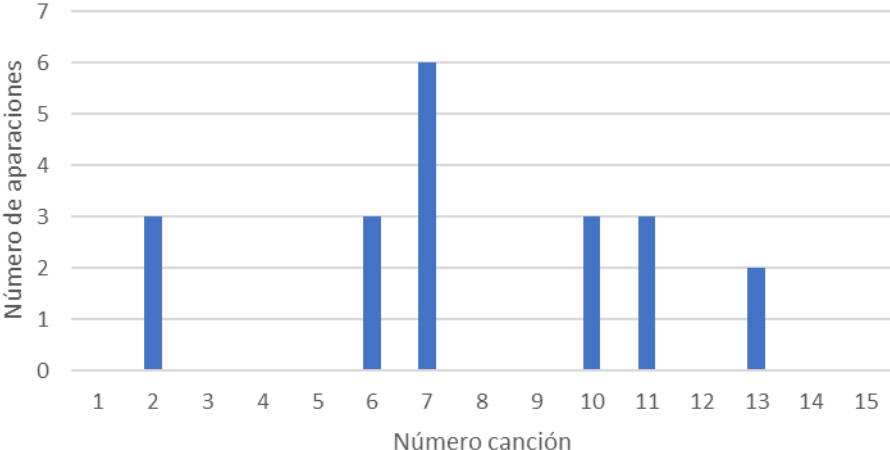
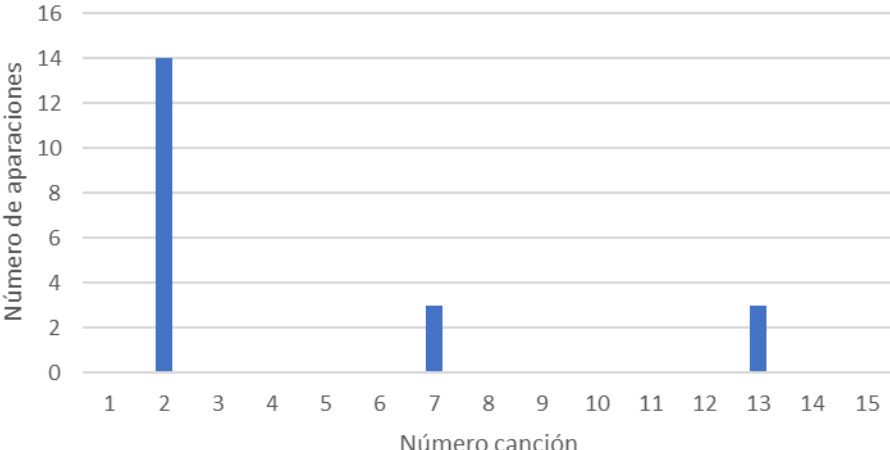
<p style="text-align: center;">Posición 13</p>  <table border="1"> <thead> <tr> <th>Número canción</th> <th>Número de apariciones</th> </tr> </thead> <tbody> <tr><td>2</td><td>2</td></tr> <tr><td>7</td><td>6</td></tr> <tr><td>10</td><td>3</td></tr> <tr><td>11</td><td>3</td></tr> <tr><td>13</td><td>6</td></tr> </tbody> </table>	Número canción	Número de apariciones	2	2	7	6	10	3	11	3	13	6	<p>La posición 13 la ocupa la canción número 7 y la número 13, “High Hopes” de Panic! At The Disco, y “Walking on Sunshine” de Katrina And The Waves.</p>		
Número canción	Número de apariciones														
2	2														
7	6														
10	3														
11	3														
13	6														
<p style="text-align: center;">Posición 14</p>  <table border="1"> <thead> <tr> <th>Número canción</th> <th>Número de apariciones</th> </tr> </thead> <tbody> <tr><td>2</td><td>3</td></tr> <tr><td>6</td><td>3</td></tr> <tr><td>7</td><td>6</td></tr> <tr><td>10</td><td>3</td></tr> <tr><td>11</td><td>3</td></tr> <tr><td>13</td><td>2</td></tr> </tbody> </table>	Número canción	Número de apariciones	2	3	6	3	7	6	10	3	11	3	13	2	<p>La posición 14 la ocupa la canción número 7, “High Hopes” de Panic! At The Disco.</p>
Número canción	Número de apariciones														
2	3														
6	3														
7	6														
10	3														
11	3														
13	2														
<p style="text-align: center;">Posición 15</p>  <table border="1"> <thead> <tr> <th>Número canción</th> <th>Número de apariciones</th> </tr> </thead> <tbody> <tr><td>2</td><td>14</td></tr> <tr><td>7</td><td>3</td></tr> <tr><td>13</td><td>3</td></tr> </tbody> </table>	Número canción	Número de apariciones	2	14	7	3	13	3	<p>La canción que los participantes han considerado más alegre es la número 2, “Hey ya!” de Outkast.</p>						
Número canción	Número de apariciones														
2	14														
7	3														
13	3														

Tabla 5 Graficas Análisis Individuales

Analizando todos datos anteriores se puede generar una gráfica con los resultados de las canciones según lo alegres que sean para el oyente.

El ranking resultante sería el siguiente (Hay posiciones con más de una canción, ya que aparecen el mismo número de veces en esa posición).

Ranking canciones

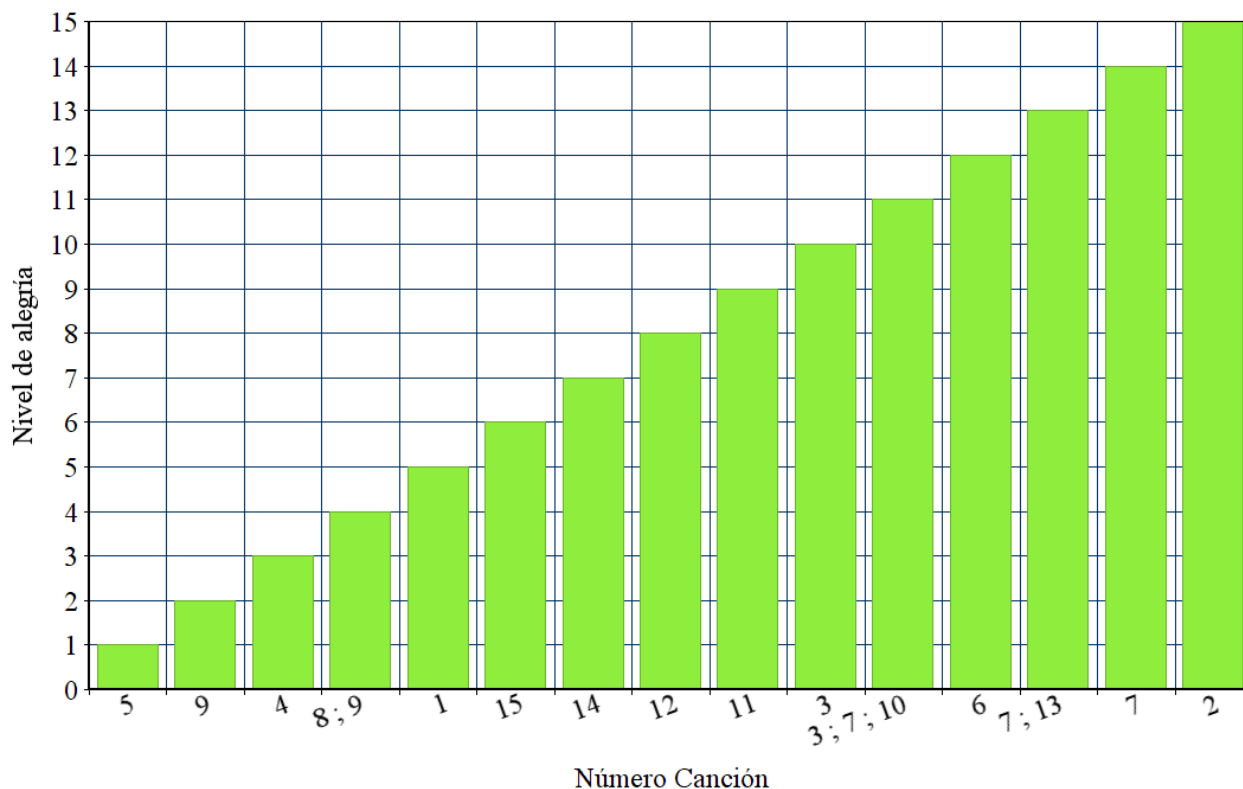


Figura 36 Ranking Canciones

Ahora que se sabe cómo ordenar cada una de las 15 canciones que se han propuesto a nuestros participantes, se va a proceder a llevar un estudio de sus características para poder estudiar de forma más concreta que elementos poseen que influyen en su calificación como canciones tristes o alegres.

En un primer momento se realiza un análisis espectral de las diferentes canciones para así recabar diferentes características básicas como pueden ser las intensidades máximas y en que frecuencias se encuentran, así como la intensidad media de la canción, etc. Por otro lado, se utilizarán otras herramientas para medir el ritmo de cada canción, y conseguir también su BPM. Estos BPM han sido analizados comparando varias páginas que se dedican a este fin, como son:

- getsongbpm
- tunebat
- songbpm
- runhundred

Los resultados de los BPM que tienen las canciones analizadas son los siguientes (ordenadas de triste a alegre):

Puesto	Nombre (autor)	BPM	Número Canción
1	The last goodbye (billy boyd)	92	5
2	Palace (Sam Smith)	81	9
3	I can feel a hot one (manchester Orchestra)	124	4
4	The end of Love (Florence + the machine)	62	8
	Palace (Sam Smith)	81	9
5	Let her go (passenger)	75	1
6	Roundabouts (michael patrick kelly)	97	15
7	Tired of waiting for you (the kinks)	122	14
8	Cats in the Cradle (harry chapin)	78	12
9	dont stop me now (queen)	156	11
10	you get what you give	114	3
11	you get what you give	114	3
	High Hopes (panic! at the disco)	82	7
	Eat Sleep Love you repeat	206	10
12	Feel good inc	139	6
13	High Hopes (panic! at the disco)	82	7
	walking on sunshine	111	13
14	High Hopes (panic! at the disco)	82	7
15	Hey ya! (outkast)	150	2

Tabla 6 BPM

Hay que recordar que hay puestos en los que aparece mas de una sola canción (como los puestos 4,11,13)

Analizando los datos que se exponen en la tabla anterior, se puede apreciar que el BPM de la canción considerada más triste es de 92 latidos por minuto, mientras que la canción considerada como más alegre tiene 150 laditos por minuto.

En la siguiente gráfica se muestran los diferentes BPM junto con su línea de tendencia. En ella, se puede apreciar que hay cierta relación entre que la canción transmita una mayor sensación de alegría y con que tenga un ritmo más rápido.

Sin embargo, se puede apreciar también que el BPM no es la única característica que influye en el sentimiento final de una melodía, ya que se pueden ver picos que se salen de la tendencia natural de la gráfica, como es el caso de la canción “Eat Sleep Love you repeat” (206 bpm). Los participantes han colocado esta canción en el puesto 11, quedándose así por

detrás de otras canciones con un BPM mucho menor, como puede llegar a ser 111 latidos por minuto, o incluso 82 latidos por minuto.

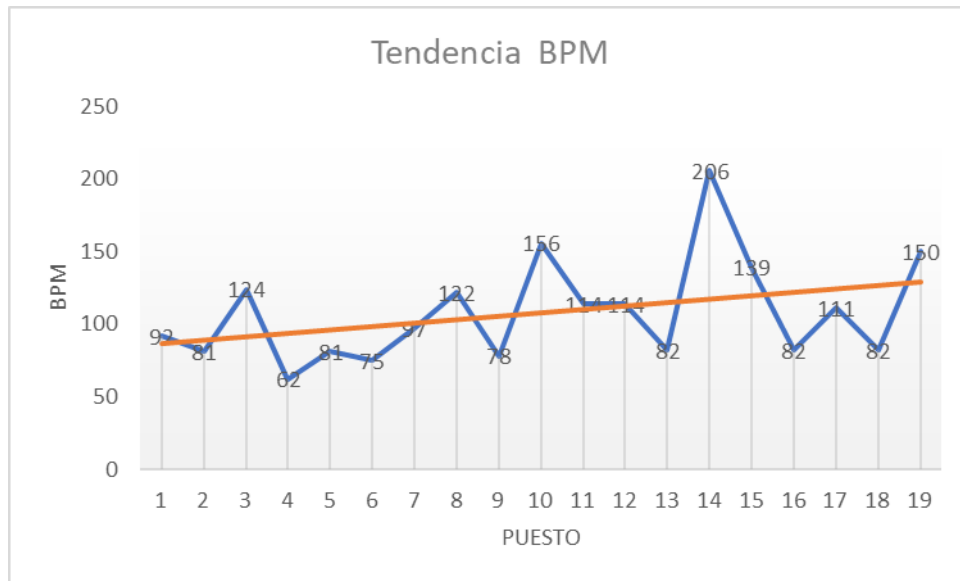


Figura 37 Estudio Ritmo

Tras haber analizado el ritmo de cada una de las canciones, se puede observar de que no se trata de una característica tan influyente como se pensaba en un principio. Por esta razón, hay que continuar el estudio de las características distintivas de cada canción de una forma mas profunda, con el objetivo de encontrar otros metodos para identificar las diferencias en los sentimientos que genera cada canción. Para ello, se va a realizar un analisis en el dominio temporal y en el dominio frecuencial. Para ello vamos a hacer uso de la tranformada de Fourier explicada anteriormente, y de varios espectrogramas o sonogramas. Para entender mejor el funcionamiento de estos gráficos y de en que consisten ambos analisis, se recomienda apoyarse en la información escrita en el Anexo E.

A continuación se va a mostrar el analisis en el dominio temporal (con 5 segundos de canción será suficiente), junto con el espectrograma o sonograma de algunas de las canciones con mas diferencia entre ellas, para intentar percibir algun patrón. Será suficiente con mostrar el de las dos o tres canciones más alegres y las dos o tres más tristes.

Los espectrogramas los serán ampliados de tal forma que quede visible la región entre 0Hz y unos 4100Hz, ya que como se ha mencionado anteriormente, en la musica se utilizan las frecuencias entre 27Hz y unos 4186 Hz.

Cada canción tendrá dos imágenes, la primera para el dominio temporal y la segunda para el espectrograma

Canciones alegres:

Canción número 2 – Canción considerada como la más alegre del ranking.

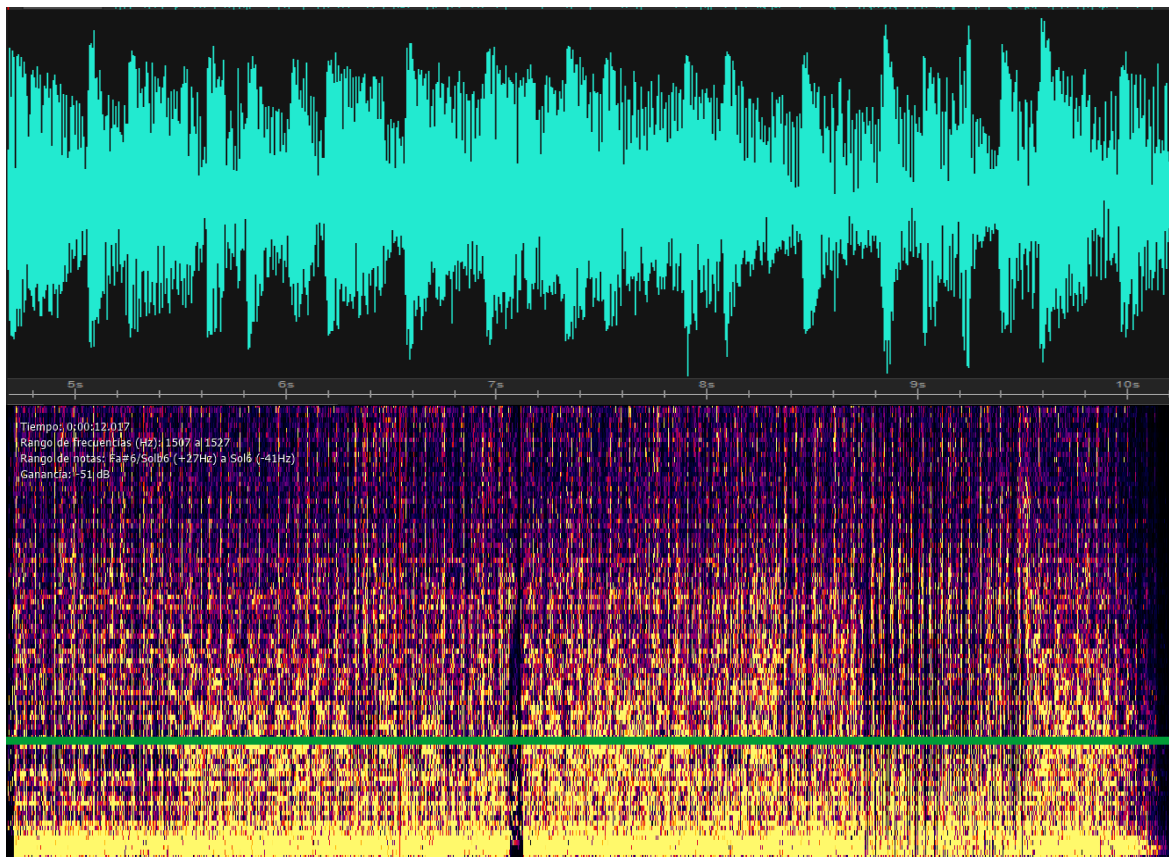


Figura 38 Estudio Canción 2

Ya que el tamaño de las imágenes es demasiado grande para que se puedan apreciar las medidas que salen en la parte superior izquierda del espectrograma, se ha colocado una línea verde de forma horizontal que marca el límite de los 1000 Hz. Todo lo que se encuentre por debajo de esa línea será menor de 1000 Hz y todo lo que la supere será mayor.

De esta forma el lector puede hacerse una idea más fácilmente en la distribución de las diferentes frecuencias de una forma aproximada.

Canción número 7 – Canción considerada como la segunda más alegre del ranking.

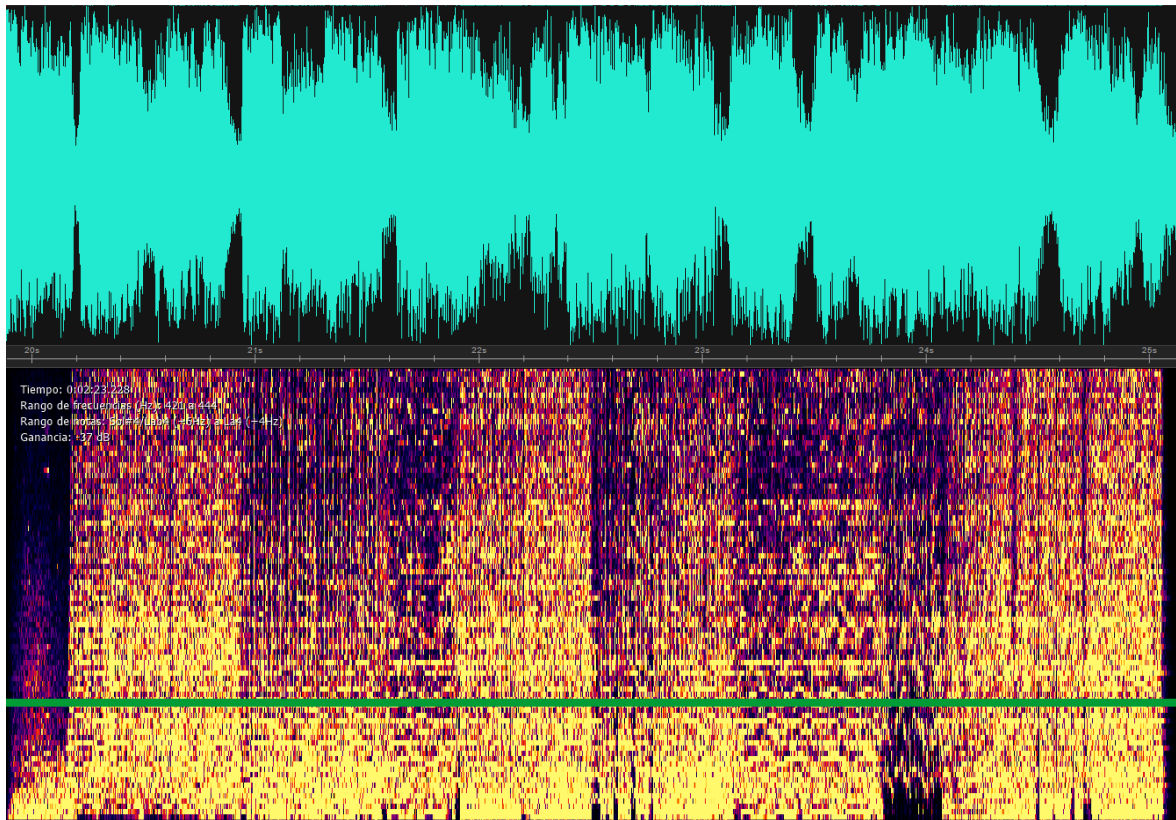


Figura 39 Estudio Cancion 7

Una vez analizadas ambas canciones y sus respectivas imágenes, podemos percatarnos de ciertas similitudes entre ellas. Por un lado, al tratarse de canciones alegres es de esperar que tengan mucha energía, y un buen número de instrumentos. Esto se puede apreciar en la primera imagen de cada una de las canciones, donde se muestra el análisis de la onda en el dominio temporal. En ambos casos se observa que se trata de dos canciones con una alta intensidad (apreciable gracias a el ancho de la onda verticalmente) y sin muchas fluctuaciones en cuanto a esta. Esto quiere decir que los momentos en los que el sonido es mas suave o mas tenue, son escasos. Con esta información se puede dar más peso a ambas hipótesis mencionadas anteriormente. Por un lado, son canciones mas alegres, por lo que tienen un mayor nivel de intensidad sonora, y por otro, tienen gran cantidad de instrumentos, por lo que se producen pocos silencios o momentos “calmados”, mas representativos de las canciones mas tranquilas.

En la segunda imagen de cada canción se puede ver el espectrograma. Analizando este tipo de representación gráfica, se puede observar que son mayormente de color amarillo. Recordar que en este tipo de gráficos, la intensidad del color depende de la intensidad de la frecuencia a la que representa, y cuanto menos intensidad, mas negro se tornará el color en este punto. En nuestro caso, se observa que hay una gran intensidad en un gran abanico de frecuencias (las frecuencias se distribuyen en el eje vertical), habiendo mas concentración en las frecuencias mas cercanas a cero. Esto encaja con la idea de en las canciones alegres es común que haya una gran cantidad de instrumentos, ya que estos emitirían sonidos en diferentes frecuencias, ocupando así un mayor rango de estas. Por ejemplo, mientras que en el piano (según la nota que toques), puedes emitir sonidos en el intervalo 27Hz-4186Hz, en la flauta es de 261Hz-2349Hz, y en la trompeta es de 165Hz-988Hz.

Habiendo analizado las dos canciones consideradas como más alegres del ranking, se plantean las siguientes ideas que se deben confirmar al estudiar las canciones consideradas como más tristes:

- Las canciones alegres tienen un nivel de intensidad sonora alto. (un volumen más elevado).
- Las canciones alegres no contienen una gran cantidad de “silencios”, o momentos en los que la intensidad disminuya bruscamente durante un intervalo de tiempo.
- Las canciones alegres contienen un gran abanico de frecuencias con gran intensidad.

Ha continuación se va a proceder a realizar el mismo procedimiento con las dos canciones consideradas como más tristes en el ranking. Estas canciones son la número 5 (**The last goodbye** - Billy Boyd) y la número 9 (**Palace** - Sam Smith). Primero se obtendrán las características comunes entre ambas canciones con el objetivo de identificar los rasgos característicos de una canción considerada como triste, y posteriormente se compararán estas características con las obtenidas anteriormente de las canciones alegres.

Canciones tristes:

Canción número 5 – Canción considerada como la 1 más triste del ranking.

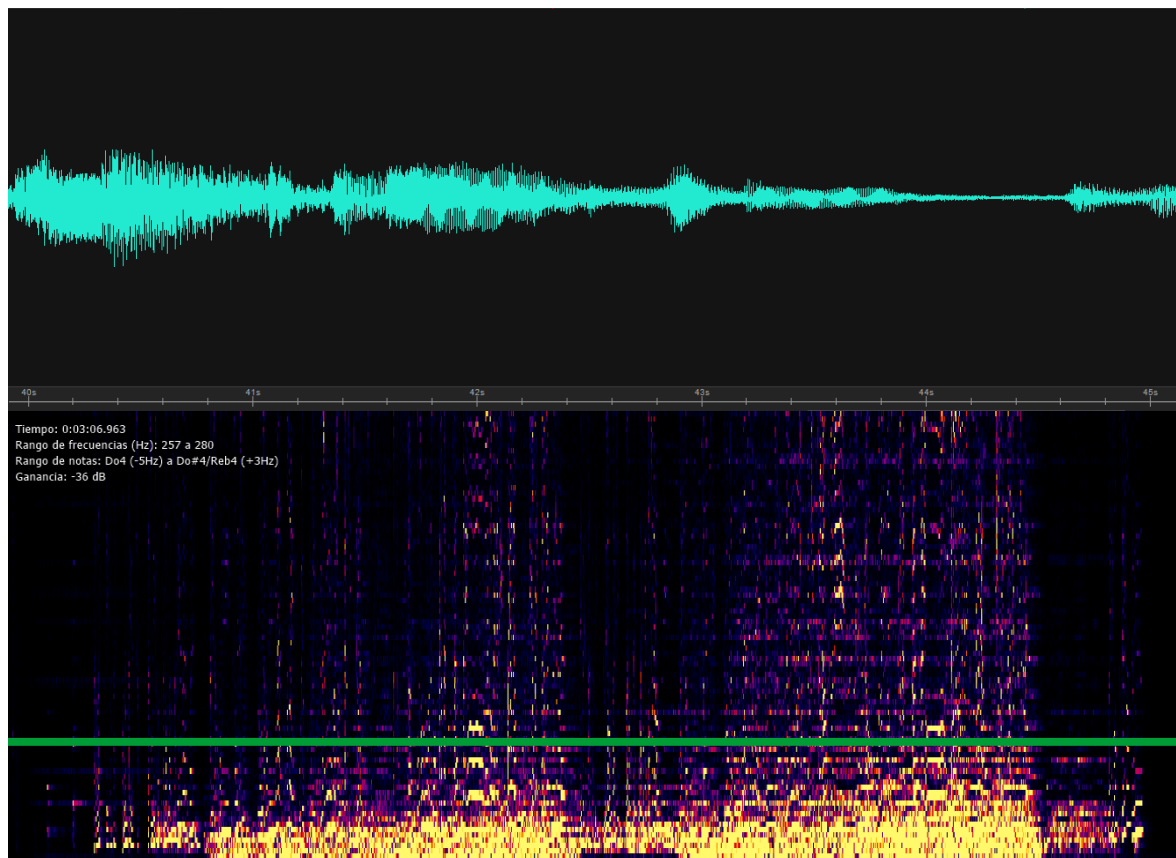


Figura 40 Estudio Canción 5

Canción número 9 – Canción considerada como la 2 más triste del ranking.

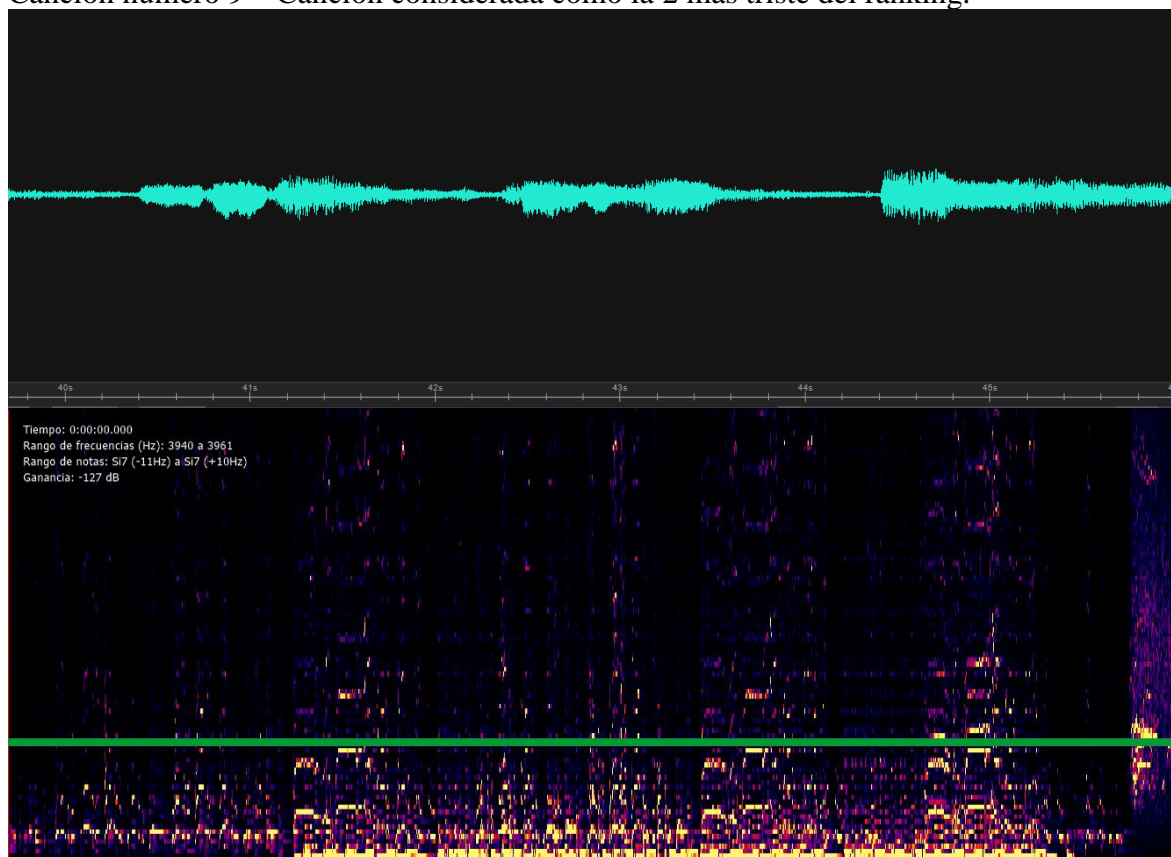


Figura 41 Estudio Canción 9

Se puede ver claramente la gran disimilitud entre las imágenes correspondientes a las canciones tristes y las analizadas anteriormente de las imágenes alegres. Por un lado, se tratan de canciones con mucha menos energía. Son canciones más tranquilas donde la intensidad es mucho menor, transmitiendo un sentimiento de “calma”. Esto se puede apreciar mejor en la siguiente imagen, donde se comparan las dos canciones tristes actuales, con la canción 7, analizada anteriormente.

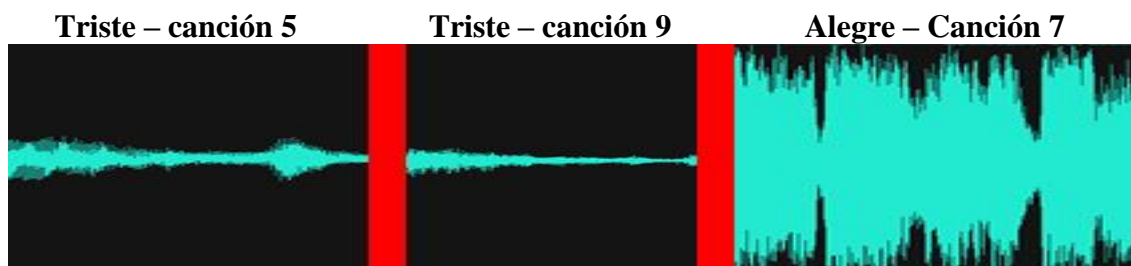


Figura 42 Comparación 1

También se puede apreciar que los silencios son mucho más representativos de las canciones tristes. El cantante suele cantar de forma más suave, sin elevar la voz. Esto se acentúa más al tener en cuenta que este tipo de canciones se caracteriza también por el bajo número de instrumentos que las componen. Es común escuchar únicamente un sonido vocal acompañado de un número reducido de instrumentos, incluso de uno solo. Por esta

razón se producen estos silencios mucho más largos en las canciones 5 o 9, en comparación con la canción 7.

Por otra parte, se puede observar que en comparación con las canciones alegres (en las cuales había un gran abanico de frecuencias) en las canciones tristes el número de frecuencias presentes es mucho menor, llegando a estar la mayoría de estas incluso por debajo de 200Hz, como sucede en la canción número 9.

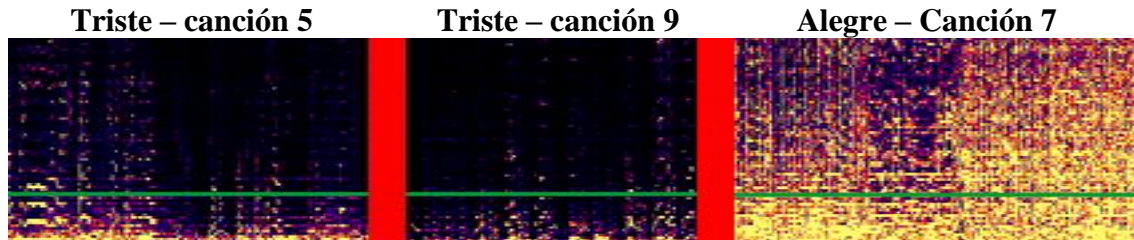


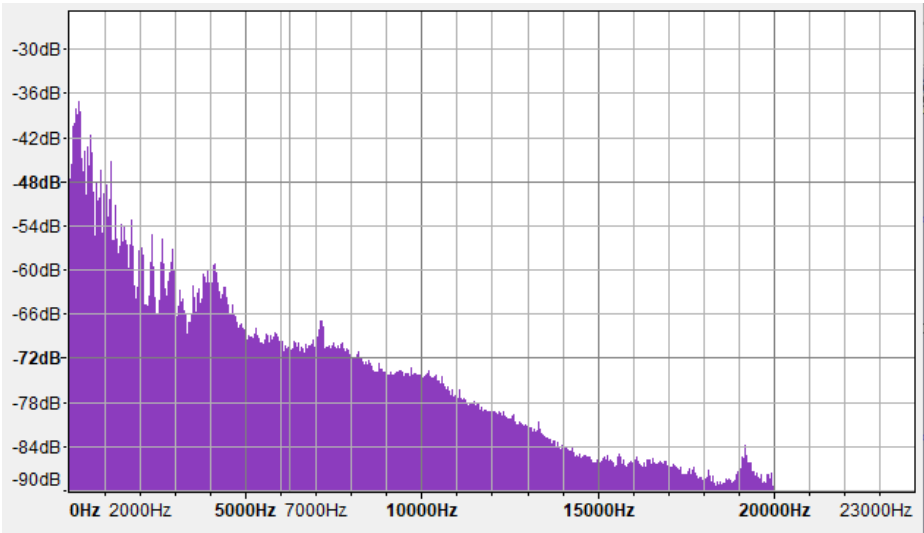
Figura 43 Comparación 2

Tras haber analizado las dos canciones consideradas como más tristes en el ranking, y habiéndolas comparado con las dos canciones consideradas como más alegres, se plantean las siguientes características propias de una canción que pretende transmitir un sentimiento de tristeza:

- Se caracterizan por un nivel de volumen reducido, transmitiendo poca intensidad sonora.
- Poseen silencios largos en los que prácticamente no se produce sonido alguno.
- Poseen un abanico de frecuencias reducido, tendiendo estas hacia las frecuencias más bajas.

Si se procede a realizar la transformada discreta de Fourier en las canciones analizadas anteriormente, se podrá observar la distribución de las frecuencias presentes, de una forma diferente a la representación en el espectrograma o sonograma. No se va a realizar un análisis exhaustivo como anteriormente, sino que se presentan estas imágenes para que el lector pueda ver de una forma diferente las diversas frecuencias que aparecen en cada canción y pueda advertir que en las canciones tristes hay una tendencia mayor hacia frecuencias bajas frente a las canciones alegres, donde las frecuencias se encuentran más distribuidas. También se puede apreciar claramente que en las canciones tristes el nivel de intensidad es menor, llegando a unos -36db en la canción 5, o -42db en la canción 9 frente a los -30db de la canción 2 o los casi -24db de la canción 7.

Canción 5
(Triste)



Canción 9
(Triste)

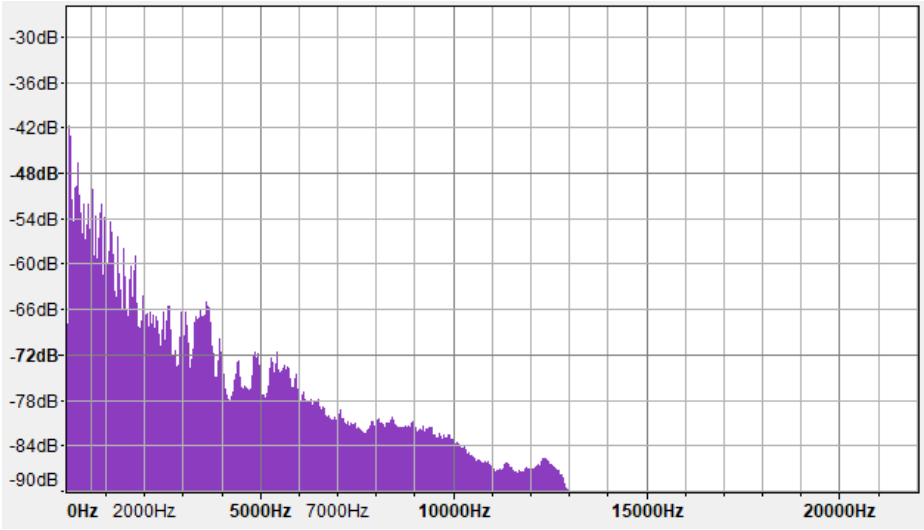
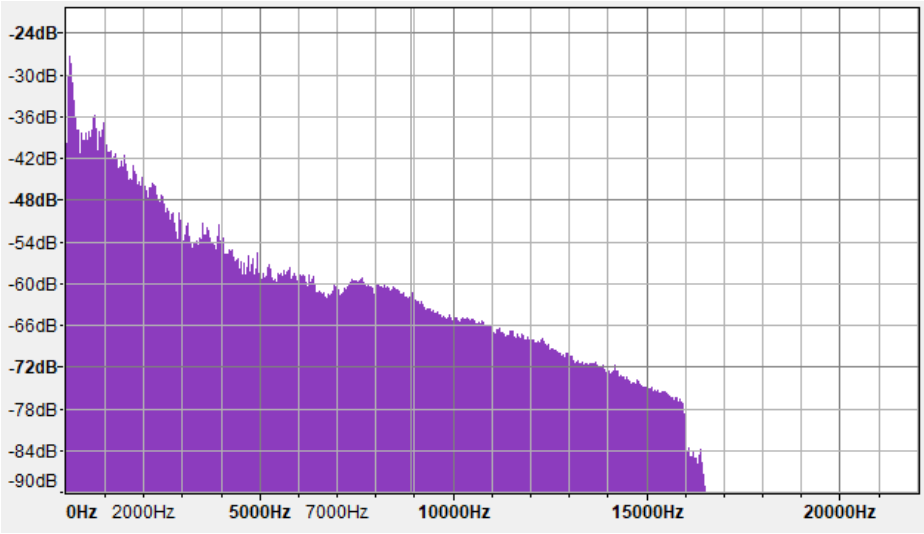


Tabla 7 Fourier Canciones Tristes

Canción 2
(Alegre)



Canción 7
(Alegre)

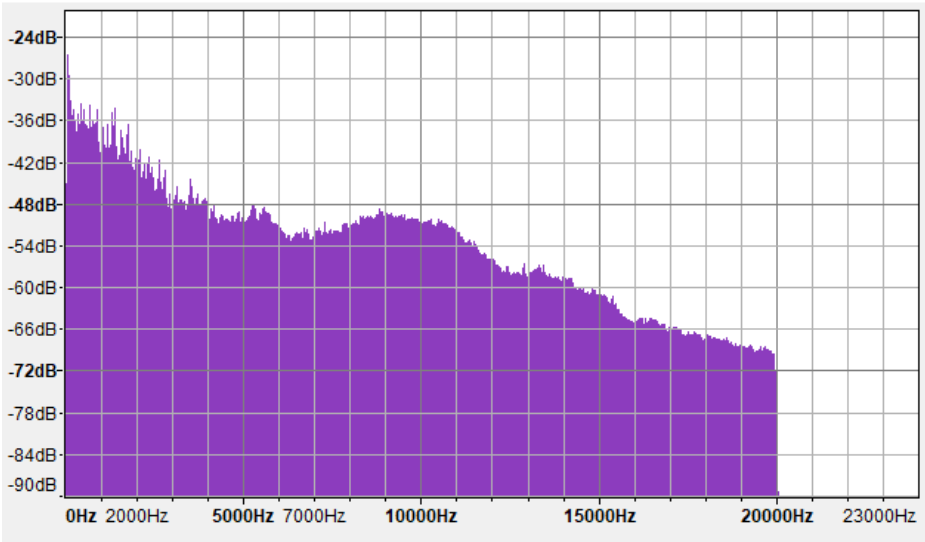


Tabla 8 Fourier Canciones Alegres

Tras la realización el sondeo, como el posterior estudio de los resultados, se llega a ciertas conclusiones con las que se podrá comenzar a realizar el desarrollo de los diferentes algoritmos, teniendo en cuenta los factores comentados en este apartado. Las conclusiones aquí mencionadas se tratan de una aproximación inicial a la solución final, y se irán analizando más detenidamente durante el desarrollo de la solución definitiva, proponiendo si se considera oportuno diversos cambios o mejoras que aporten al algoritmo una mayor precisión a la hora de realizar el análisis de cualquier canción.

8.4 Representación gráfica de una señal de audio

En esta sección se va a mostrar alguno de los procedimientos con los que se puede extraer diferentes datos de una señal de audio, y que servirán de apoyo a la hora de realizar diferentes análisis con el fin de desarrollar la solución del presente trabajo.

Mediante estos procedimientos basados en el análisis del dominio espectral y temporal de una señal de audio, se podrá observar de una forma gráfica la configuración de la que se caracteriza una canción dependiendo del sentimiento que se quiera generar con ella.

8.4.1 Dominio frecuencial

Este tipo de análisis se emplea para representar una señal respecto a la frecuencia. En este tipo de análisis se podrá observar la cantidad o intensidad de esa señal en el rango de frecuencias, y de esta forma, ver cómo está distribuida.

El análisis en el dominio frecuencial permite observar una señal desde otra perspectiva, que en muchos aspectos facilita su comprensión. Ciertas características se pueden apreciar más fácilmente desde el dominio frecuencial.

En un caso ideal, una señal estaría compuesta únicamente por una sola componente frecuencial. Las señales de audio que se estudiarán a lo largo del presente proyecto no se tratarán de casos ideales, y estarán compuestas de diferentes compuestos, los cuales, no serán tan sencillos de analizar.

En la representación frecuencial de una señal se pueden encontrar distintas metodologías a la hora de mostrar los resultados obtenidos. En este caso se usará la forma más asentada o más comúnmente utilizada, que consiste en la representación de cada componente frecuencial en la línea del eje horizontal, mientras que la longitud de cada línea en el eje vertical representará la potencia de cada una de esas componentes. De esta forma se podrá observar cada una de las amplitudes pertenecientes a las frecuencias presentes en la señal que ha sido analizada.

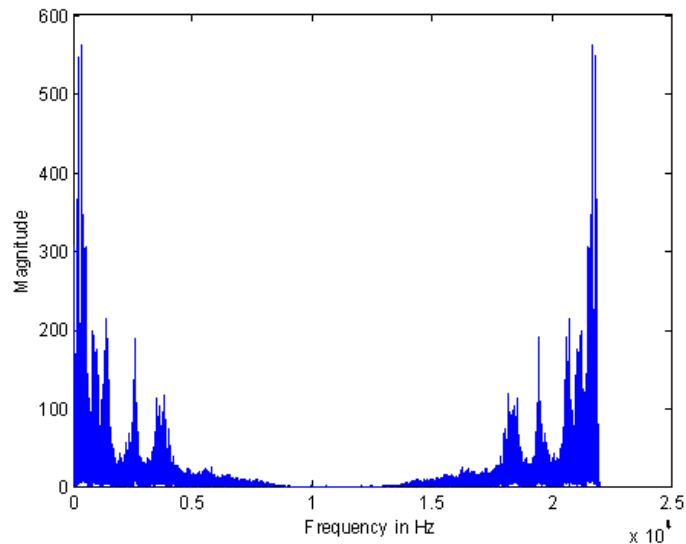


Figura 44 Componentes Frecuenciales

Gracias a este tipo de representación de una señal respecto a la frecuencia, se puede obtener diferente información que no podría aportar, por ejemplo, un oscilograma. De esta forma permite estudiar ciertas características de la señal como es el timbre, la sonoridad, o la amplitud de cada frecuencia, que aporta una mayor comprensión de las propiedades musicales del audio [Referencia 26].

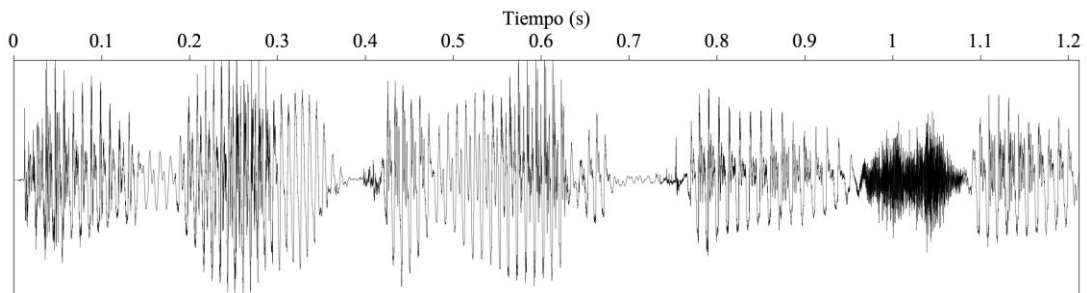


Figura 45 Oscilograma

Como se ha mencionado anteriormente, una muestra de sonido rara vez está compuesta de una única onda, si no que suele estar acompañada de multitud de otras ondas cuyas frecuencias tienen amplitudes menores que la de la onda principal o fundamental y se denominan parciales. De hecho, en multitud de sonidos no se puede saber con seguridad cuál de las ondas se trata de la fundamental y cuales son parciales que acompañan a esa onda.

El método usado para transformar cualquier muestra al dominio frecuencial para su posterior representación, es un proceso matemático llamado “transformada de Fourier”. Con este método se puede descomponer la señal de audio en las diferentes señales más simples que la componen, es decir, separar la onda en las componentes de frecuencia que la forman. Al realizar este método, lo que se obtiene son las partes reales e imaginarias de las ondas pertenecientes a cada una de las frecuencias, por lo que habría que realizar el valor absoluto a la transformada de Fourier para lograr saber la energía que contiene cada una de las frecuencias.

8.4.2 Dominio temporal

Mientras que en el dominio frecuencial se ha podido observar las características de una señal respecto a las frecuencias que las componen, en el dominio se analizará esa señal y sus cambios a lo largo del tiempo. A este tipo de análisis también se le conoce como oscilograma.

Con este método de representación de una señal se puede ver cómo evoluciona su amplitud a lo largo del tiempo, pudiendo observar si esta disminuye o aumenta en cada instante. Gracias a esto se podrá realizar un estudio más detallado de una onda, pudiendo observar la estructura que la caracteriza, sus variaciones y la duración de estas, y su sonoridad.

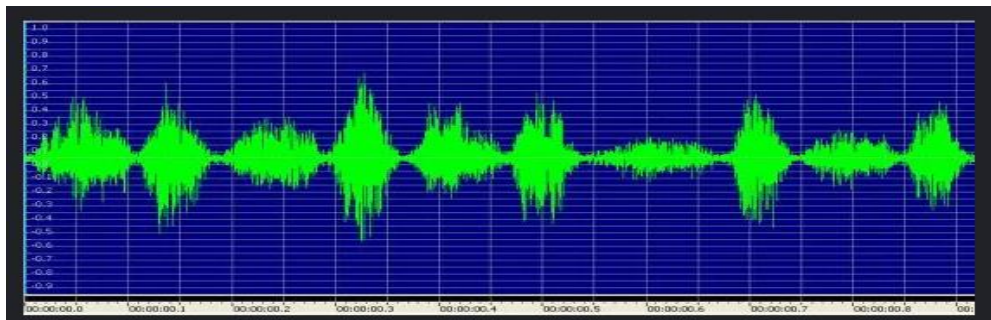


Figura 46 Oscilograma. Eje vertical(amplitud), Eje horizontal (tiempo)

Este tipo de gráficos se representan entorno a un punto central (cero), que se trata del valor medio de la presión en la onda, y va en aumento desde ese punto según la amplitud que tenga la señal en ese instante.

8.4.3 Espectrograma o sonograma

A diferencia de las gráficas que se han visto hasta ahora, donde se representaba el espectro de una sola ventana, en el espectrograma se irá un poco más lejos, y se calculará el espectro de tramas enventanadas de una señal. Como resultado se tendrá una gráfica donde, mediante el color, se representará la energía que contiene una determinada frecuencia, y como esta varía con el tiempo. De esta forma quedará una gráfica con 3 dimensiones, a diferencia de las que se tenía hasta ahora, que eran de 2.

El espectrograma o sonograma, resultará muy útil para poder comprender mejor la estructura de cada canción, ya que no solo se podrá ver las frecuencias que contienen a lo largo del tiempo y su evolución, sino que también estarán representadas sus amplitudes mediante colores [Referencia 27].

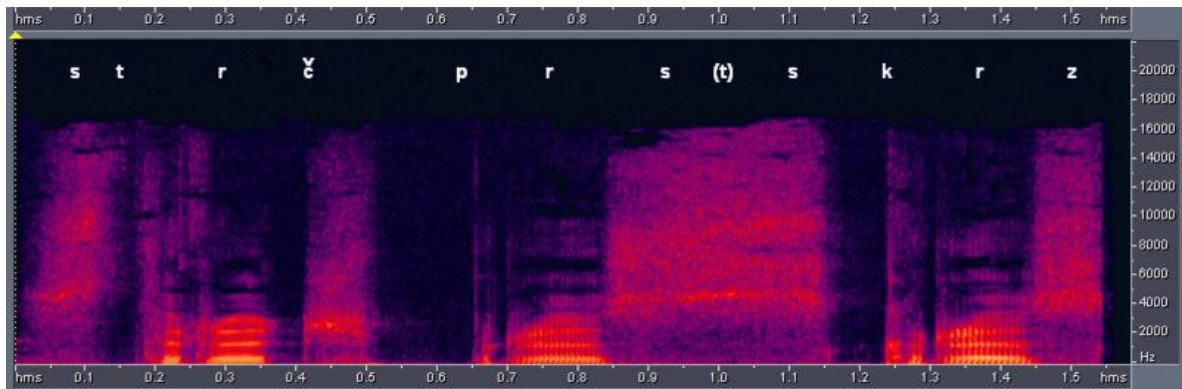


Figura 47 Espectrograma

Para poder realizar un espectrograma hay que buscar un modo de poder procesar las señales en momentos de tiempo determinados. Esto se debe a que, a diferencia de la teoría, donde se manejaban señales ideales, infinitas y estacionarias, en la práctica, habrá que manejar señales finitas, y no estacionarias. Para poder procesar este tipo de señales se hará uso de una ventana temporal de un tamaño concreto, de esta forma, se procesará un número limitado de muestras.

“Las **ventanas** son funciones matemáticas usadas con frecuencia en el análisis y el procesamiento de señales para evitar las discontinuidades al principio y al final de los bloques analizados.” – Wikipedia.

Con los datos analizados en esa ventana se calculará su contenido frecuencia, y posteriormente, los datos serán representados en una gráfica de tres dimensiones.

Este procedimiento se conoce como STFT o Transformada de Fourier de corta duración, y consiste en dividir una señal en pequeños fragmentos conocidos como tramas de análisis. Los pasos que habría que hacer para llevar a cabo el STFT son los siguientes:

1. Coger una muestra de la señal y aplicar la función ventana para poder llevar a cabo la Transformada de Fourier en ese intervalo.
2. Se mueve la función ventana a la siguiente muestra de la señal. Para realizar esto de manera correcta hay que realizar un solapamiento, para que ambos intervalos se superpongan ligeramente y de esta forma que produzca una continuidad temporal. Una vez hecho esto, se aplica la Transformada de Fourier a la nueva muestra capturada.
3. Se continua este proceso hasta haber analizado toda la señal total.

Una vez que se han obtenido todas las trazas necesarias hasta completar la señal, se indexan en una matriz. En esta matriz se podrán observar las diferentes variaciones de espectro y su energía en cada uno de los diferentes intervalos analizados durante toda la señal.

Una de las formas de representación de un espectrograma es la siguiente:

- Eje horizontal: Línea de tiempo.
- Eje vertical: Frecuencia expresada en Hz.

- Representada por una gama de colores: Energía en dB. Las intensidades mayores suelen ser representadas en colores más brillantes o intensos, a excepción de si está representado en la escala de grises. En esta escala los valores mayores serán representados por colores más oscuros, y los menores con colores más claros.

Gracias a este tipo de representación o procesamiento de una señal, se podrán aprovechar las diferentes propiedades relacionadas con la intensidad de la energía en los gráficos de dos dimensiones vistos anteriormente, frecuencia y tiempo, en vez de aprovechar solo una, tiempo o frecuencia. El único problema es que cuando se fija una función ventana, si se quiere aumentar la resolución temporal, solo se puede mejorar a expensas de empeorar la resolución que tiene la frecuencia (y viceversa). Esto quiere decir que ambas resoluciones de estas propiedades son inversamente proporcionales.

8.5 Conocimientos previos al diseño

8.5.1 La música

Hay multitud de definiciones distintas sobre que es la música, pero se puede decir que se trata del arte de organizar de forma lógica y sensible una combinación de diferentes sonidos y silencios, y de esta forma, conseguir crear una emoción determinada en el oyente. [Referencia 4]

La música, como cualquier otra forma de expresión artística, se trata de un producto de carácter cultural, y puede tener como objetivo varias funciones como pueden ser el entretenimiento, la comunicación, la ambientación etc.

8.5.2 El sonido

Desde el punto de vista de la física, un sonido consiste en un fenómeno asociado con la propagación de una onda por un medio elástico, lo que produce la vibración de un cuerpo. En la propagación de este fenómeno, lo que se transporta es energía, no materia.

Para que este proceso tenga lugar, es necesaria la existencia de un cuerpo que genere la vibración, llamado “foco”, y de un cuerpo elástico por donde se transmitirán estas vibraciones, generando lo que se denomina como onda sonora. [Referencia 5]

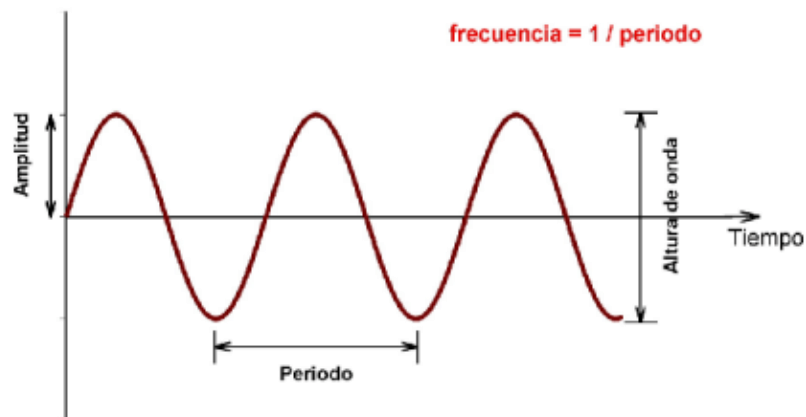


Figura 48 Onda

Se pueden diferenciar unos sonidos de otros comparándolos por las características que tienen en cuanto a la forma en que los percibimos. Estas características son las siguientes: el tono (número de vibraciones por segundo del foco), timbre (nos permite distinguir dos ondas sonoras que tienen diferentes fuentes sonoras), intensidad (fuerza con la que percibimos un determinado sonido). Se procederá a explicar estas características con más detalle en las siguientes secciones.

En ciertos sonidos, como los producidos por los diversos instrumentos musicales de cuerda, existe una repetición que somos capaces de percibir, (es lo que llamamos altura

tonal) por debajo de esta vibración, existen un conjunto de otras vibraciones a diferentes frecuencias, que son múltiplos de una frecuencia a la que llamaremos fundamental. Por ejemplo, en el caso de un violín en el que se toca la nota “La”, no solo se producirá una frecuencia a 440Hz (es la frecuencia que caracteriza la nota “La”), sino que se producirá la suma de otras series de vibraciones sinusoidales (880Hz, 1320Hz, etc....) que son llamados “armónicos”. La suma de todos estos armónicos, da como resultante una onda que vibra de forma cíclica y que ya no es sinusoidal, y puede no ser periódica. [Referencia 6]

Los armónicos son las diferentes vibraciones y frecuencias que se encuentran componiendo un movimiento oscilador, múltiplos enteros de la frecuencia fundamental. [Referencia 10]

Esto se puede ver más claro en la siguiente imagen:

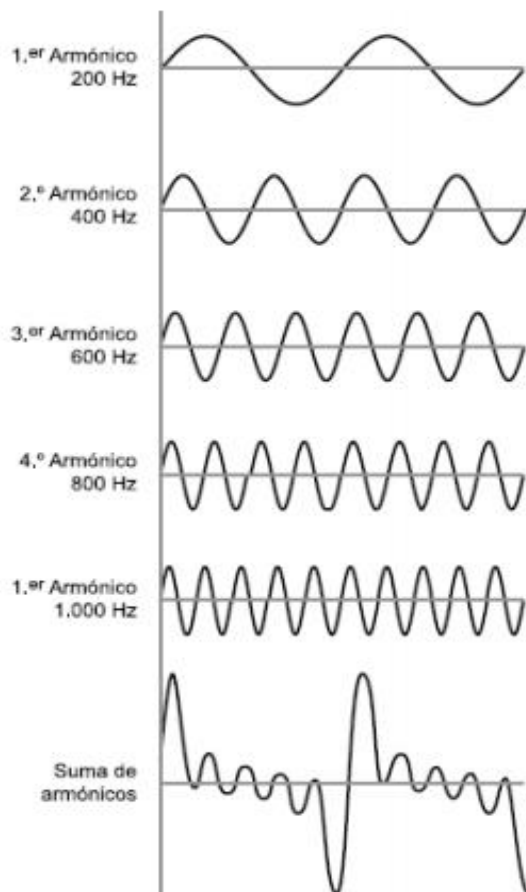


Figura 49 Armónicos [Referencia 7]

8.5.3 Frecuencia

La frecuencia de un sonido indica el número de veces que se producen vibraciones u oscilaciones completas en el cuerpo que transmite un sonido, en un periodo de tiempo de un segundo.

Sus unidades en el sistema internacional son los Hertzios (Hz) y ya que se trata del número de repeticiones que se produce una determinada vibración, o un determinado patrón por unidad de tiempo, su fórmula es la siguiente:

$$f = \frac{1}{s} = 1 \text{ Hz}$$

Los seres humanos solo podemos escuchar sonidos entre 20Hz – 20Khz, todo lo que se encuentre por debajo se considerará infrasonido, mientras todo lo que se encuentre por encima será un ultrasonido.

Los sonidos que el ser humano considera como más graves, se encuentran en las frecuencias menores, mientras que a medida que vayamos aumentando de frecuencia, los sonidos serán más agudos.

En la música no se utiliza todo el espectro audible de frecuencias que podría ser escuchado por el ser humano, se utiliza otra gama de frecuencias que va desde los 27 Hz (primera nota del piano) hasta unos 4186 Hz (última nota del piano).

Las frecuencias que se encuentran fuera de este rango no son utilizadas en la música, ya que no todas las personas son capaces de escucharlas, y aunque lo fuesen, podrían percibir estos sonidos como desagradables.

8.5.4 Amplitud

La amplitud de una onda sonora es la cantidad de energía que transporta la vibración de un sonido. Cuanto mayor energía, mayor será la vibración y más fuerte percibiremos ese sonido cuando golpee contra el tímpano. Por esta razón se podría considerar que la amplitud nos indicará lo fuerte o débil que será un sonido, y corresponderá con el valor máximo y mínimo que alcance su onda sinusoidal.

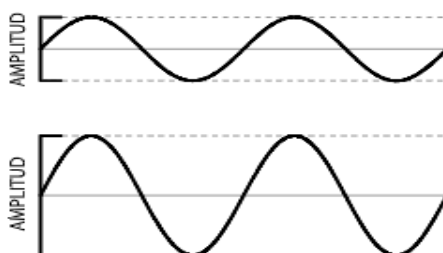


Figura 50 Amplitud

Las unidades con las que nos referimos a la amplitud de una onda son los decibelios (dB), aunque puede expresarse también en pascles o milibares. Para que un sonido sea audible por el ser humano, debe superar los 0 dB (umbral auditivo) y no superar los 140 dB (umbral del dolor).

8.5.5 El oído y el sonido

Para entender mejor cómo influye la música en los sentimientos y comprender correctamente en que consiste un análisis espectral, hay que plantearse una serie de preguntas. ¿cómo se percibe música en nuestro cuerpo? En esta sección se analizará cómo responde nuestro oído frente al sonido.

El oído es un órgano del cuerpo humano que se encuentra activo constantemente. Este se encarga de transformar las ondas sonoras recibidas en pulsos eléctricos comprensibles por el cerebro. El sonido es una onda que puede vibrar rápida o lentamente, según esta velocidad en la vibración, se producen sonidos graves (vibración lenta) o sonidos agudos (vibración rápida). Estas ondas llegarán a la membrana timpánica, que comenzará a vibrar y transmitirá esta vibración hasta el oído interno mediante el martillo, yunque y estribo (oído medio). En el oído interno podemos encontrar la cóclea, cuyo entendimiento desempeña un papel esencial para comprender el análisis espectral. Es en la cóclea donde esta energía del sonido en forma de vibraciones, se transforma en señales eléctricas. [Referencia 12]

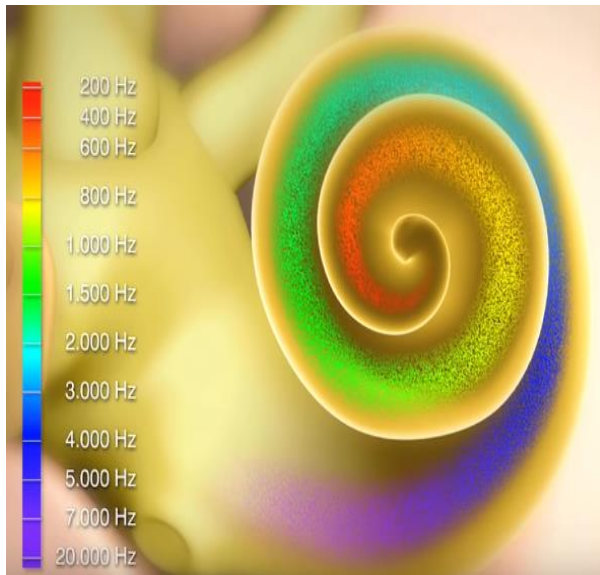


Figura 51 Oído

En esta zona del oído se encuentran dispuestas las células sensoriales o células ciliadas, las cuales cubren por completo toda extensión de la cóclea (como se puede apreciar en la imagen). Estas células tienen diferentes grados de sensibilidad para detectar diferentes tonos o frecuencias, que se pueden ver representados por los diferentes colores. Como se puede apreciar, estas células están dispuestas como si se tratase del teclado de un piano. En la zona inferior tenemos aquellas células responsables de las altas frecuencias, mientras que, en la zona superior o más interna, tenemos las bajas. [Referencia 13]

Por otro lado, se puede definir el sonido como el conjunto de ondas mecánicas que viajan a través de cualquier medio sólido, líquido o gaseoso y que producen algún tipo de perturbación o vibración que puede ser captada por el oído. Hay que diferenciarse lo que es el fenómeno físico en sí, y lo que es el fenómeno psico acústico, ya que, si no existe un oyente, se produce el evento sonoro de igual manera, pero no se produce evento auditivo alguno.

8.5.6 Aproximación a la comprensión musical

Una vez que se ha comprendido en qué consiste el sonido se va a proceder a realizar un acercamiento a una comprensión superficial de la música, que parámetros se tendrán que tener en consideración para su correcto análisis, y cuál de ellos influye en mayor o menor medida en los sentimientos que esta genera en las personas.

“El arte de organizar sensible y lógicamente una combinación coherente de sonidos y silencios respetando los principios fundamentales de la melodía, la armonía y el ritmo, mediante la intervención de complejos procesos psico anímicos.”
- Wikipedia

Primeramente, se aclararán ciertos términos respecto al sonido, los cuales se mencionarán a continuación, ya que podrán afectar de manera negativa a una correcta comprensión de esta memoria. Se realizará una explicación concisa con el simple objetivo de aclararlos:

Intensidad: Se trata de la cantidad de energía acústica que transporta el sonido. La intensidad de un sonido cualquiera viene definida por su potencia, la cual, a su vez, está definida por la amplitud de la onda que transporta. Mediante esta característica apreciamos si un sonido es grave o fuerte.

Duración: Es tiempo que dura la onda. Se entiende como el tiempo que dura la vibración del objeto emisor.

Tono: Mediante esta característica se verá si un sonido es grave, agudo, o medio. En lo que respecta al tono, hay que tener en consideración ciertas limitaciones humanas. El ser humano puede escuchar entre 20Hz y 20000Hz. Todo lo que se encuentre por debajo de esos 20Hz se considerará infrasonido. Por el contrario, todo lo que se encuentre por encima de 20000 será considerado ultrasonido.

Timbre: El timbre es lo que permite diferenciar si el sonido es producido por un instrumento, por otro, por una persona u otra. Mas concretamente permite diferenciar dos sonidos que, aunque posean las mismas características como, amplitud o duración, la onda que los caracteriza es diferente. Esto se puede apreciar mucho más fácilmente con un ejemplo. [Referencia 9]

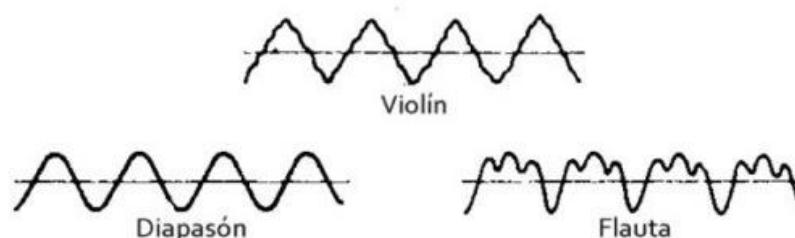


Figura 52 Caracterización Timbre

Como se puede apreciar en la imagen anterior, estas tres ondas (aunque con ciertas similitudes) son totalmente diferentes las unas de las otras. [Referencia 14]

Las reacciones emocionales a la música se pueden inducir con la combinación de las características mencionadas anteriormente, una serie de elementos con los que se puede jugar musicalmente para generar alegría, miedo, tristeza, ira, sorpresa o repugnancia.

En esta memoria no nos vamos a centrar en todos estos sentimientos, ya que la línea que los separa puede llegar a ser muy delgada en ciertos casos y totalmente personal de cada individuo.

En el análisis emocional que produce la música se tiene que tener en consideración que la cultura del oyente es esencial y totalmente determinante en el sentimiento que una pieza musical genera en él. Se trata de la característica que más va a influir de manera decisiva comparada con cualquier otra variable en las emociones que le sugiere la melodía a cada persona. Por esta sencilla razón no existen parámetros universales, o una fórmula perfecta sobre la emoción concreta que puede transmitir una determinada obra musical. Estas emociones no dejan de ser más que una convención que cada uno ha establecido dentro de su cultura. [Referencia 15]

Por exponer un ejemplo, si el lector ha viajado alguna vez al extranjero y en algún momento decidió disfrutar de la música poniendo una emisora extranjera, se ha podido dar cuenta de que no llega a conectar con ella del todo (por supuesto, este ejemplo trata sobre música nacional del país en cuestión). Si, por ejemplo, el lector mantiene una escucha más repetitiva de ese tipo de música, para acostumbrarse a ella, vera que aprende a “apreciarla”, ya que adquiere una mayor comprensión de los patrones que esta encierra.

Por supuesto hay patrones comunes en varias culturas, características de la música que tanto en uno como en otro lugar despiertan cierta emoción. Por ejemplo, en el 2004 se realizó un estudio en el que se pudo asociar el sentimiento de la ira con el volumen, tanto en la música asiática como en la occidental. [Referencia 16]

También es importante mencionar la importancia del modo que tenga la canción. Para aclarar de que se trata el modo, hay que conocer el concepto de acorde. Un acorde consiste en hacer sonar 3 o más notas a la vez. Hay varios tipos de acorde, como mayor, menor, disminuido, etc. El tipo de acorde que se forme dependerá de la distancia que haya entre sus notas. [Referencia 17]

Por ejemplo, si se dispone de un teclado con la siguiente sucesión de notas:

Do, Do#, Re, Re#, Mi, Fa, Fa#, Sol, Sol#, La, La#

Re+Fa#+La sería un acorde en modo mayor.

Re+Fa+La sería un acorde en modo menor.

El modo menor suele estar ligado a una sensación de desagrado o tristeza, mientras que el modo mayor se asocia más a la alegría, a sensaciones activantes o enérgicas, he incluso a sensaciones de amenaza.

En un principio se considera el análisis del modo de las piezas musicales para obtener una referencia sobre el tipo de emoción que esta evoca en el oyente, pero se descarta debido a que la obtención del modo de una canción implicaría el conocimiento de las notas que la forman. Para ello ya no sería suficiente con aportar una pieza musical en un formato de audio común, como por ejemplo el “.WAV”, si no que sería necesario aportar un archivo en un formato que contuviese las notas, como por ejemplo un “. MIDI”. Por la dificultad que entraña este desarrollo y ya que no se desea que se tengan que aportar las notas de la canción, se descarta esta implementación.

En general, podríamos sacar las siguientes conclusiones respecto a la música occidental.

Alegría	Se caracteriza por un tiempo rápido, melodías constantes y potentes en cuanto a la intensidad.
Tristeza	Se caracteriza por un tiempo más lento, suelen ser melodías más disonantes y con una intensidad más débil respecto a las anteriores.

A continuación se exponen algunos ejemplos. Se analiza una muestra de la canción “Dont stop me now” de la banda Queen. Su bpm (beats per minute (pulsos por minuto)) es de 150, tiene una tonalidad mayor, gran energía en los diferentes instrumentos. En este caso se trata de canción alegre y motivacional. Sin embargo, “Wake me up when september Ends” del grupo Green Day se caracteriza por ser una canción triste, con un bpm de únicamente 105, y una intensidad mucho menor en sus instrumentos.

Por otro lado, aunque un timbre suave se asocia con ternura o tristeza, y un timbre agudo se suele asociar con emociones más intensas como la ira, no suele ser un factor tan influyente en el estado emocional de la canción. Depende más de la música que se toque y de cómo se ejecute. [Referencia 18]

Hay que tener en cuenta que todo lo mencionado anteriormente son generalizaciones basándose en la música occidental, y que también hay que tener en cuenta que una canción no tiene por qué seguir siempre estas reglas, ni si quiera en el transcurso de una única canción [Referencia 19]. Una canción puede tener diferentes ritmos en ella, mezclando diferentes tonalidades en distintas partes de la misma, y cambiando intensidades e instrumentos, por ejemplo, en el estribillo.

También es importante recordar que, como se ha expuesto con anterioridad, el estado emocional que es generado por una determinada canción se ve influido en su mayoría por la cultura de la persona que la escucha, y si bien se pueden sacar ciertas conclusiones basándose en el “sentido común” de las composiciones y los estilos actuales en nuestra cultura, no sería una conclusión tan precisa o general como se pretendiera. Por esta razón se realiza un pequeño sondeo con la participación 20 personas sobre 15 canciones diferentes. En este sondeo se les ha pedido que ordenen esas canciones según el estado inicial que genere en ellos, numerándolas progresivamente desde la canción más triste a la canción más alegre.

Este sondeo se puede ver en el anexo 8.3, con sus respectivos resultados.